




МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра информационных систем

Обработка потока копий электронных документов

Бакалаврская работа
09.03.02 Информационные системы и технологии
Профиль «Встраиваемые вычислительные системы и интернет вещей»

Зав. кафедрой		Борисов Д. Н.	к.т.н., доцент	<u>19.06.2025</u>
Обучающийся		Ускова А. Е.		
Руководитель		Фертиков В. В.	к.ф.-м.н., доцент	


Воронеж 2025



МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра информационных систем

УТВЕРЖДАЮ
заведующий кафедрой
информационных систем

 Д. Н. Борисов
13.09.2024

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
ОБУЧАЮЩЕГОСЯ Усковой Анны Евгеньевны

1. Тема работы «Обработка потока копий электронных документов», утверждена решением ученого совета факультета компьютерных наук от 23.10.2024
2. Направление подготовки: 09.03.02 Информационные системы и технологии
3. Срок сдачи законченной работы: 01.06.2025
4. Календарный план (строится в соответствии со структурой ВКР):

№	Структура ВКР	Сроки выполнения	Примечание
1	Введение	01.09.2024 – 02.03.2025	
2	1. Анализ предметной области	01.09.2024 – 29.02.2025	
3	1.1 Обзор существующих подходов к обработке документов	02.09.2024 – 22.02.2025	
4	1.2 Особенности работы с электронными копиями паспортов РФ	01.09.2024 – 10.12.2025	
5	1.3 Анализ существующих решений и их ограничений	04.10.2024 – 01.02.2025	
6	2. Исследование методов и технологий обработки изображений документов	01.09.2024 – 07.03.2025	
7	2.1 Классические методы компьютерного зрения	05.09.2024 – 10.12.2024	
8	2.2 Применение нейронных сетей для детекции объектов	28.01.2025 – 21.02.2025	
9	2.3 Технологии оптического распознавания символов	01.09.2024 – 10.12.2025	
10	2.4 Сравнительный анализ методов определения ориентации	02.11.2024 – 07.03.2025	
11	3. Разработка системы обработки паспортов	03.09.2024 – 21.05.2025	

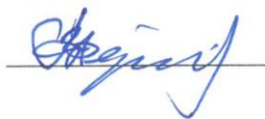
12	3.1 Исследование и анализ подходов к обработке документов	10.09.2024 – 07.03.2025	
13	3.2 Проектирование архитектуры системы	01.12.2024 – 24.03.2025	
14	3.3 Подготовка данных и обучение моделей машинного обучения	10.03.2025 – 25.03.2025	
15	3.4 Программная реализация модулей обработки	25.03.2025 – 18.04.2025	
16	3.5 Экспериментальное исследование и тестирование системы	19.04.2025 – 01.05.2025	
17	3.6 Оценка эффективности разработанной системы	02.05.2025 – 21.05.2025	
18	Заключение	22.05.2025 – 23.05.2025	
19	Список использованных источников	15.11.2024 – 28.05.2025	
20	Приложения	22.05.2025 – 29.05.2025	

Обучающийся



Ускова А. Е.

Руководитель



Фертиков В. В.

РЕФЕРАТ

Бакалаврская работа 90 с., 9 рис., 3 табл., 26 использованных источников, 1 приложение.

ОБРАБОТКА ДОКУМЕНТОВ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, YOLO, ОПТИЧЕСКОЕ РАСПОЗНАВАНИЕ СИМВОЛОВ, ОБНАРУЖЕНИЕ ОБЪЕКТОВ, ПАСПОРТ РФ, НЕЙРОННЫЕ СЕТИ, PYTHON

Объект исследования – процесс автоматической обработки электронных копий паспортов граждан Российской Федерации.

Цель работы – разработка информационной системы для автоматизированной обработки потока копий электронных документов с применением современных методов компьютерного зрения и машинного обучения.

В процессе выполнения работы проведено исследование существующих подходов к обработке документов, выполнен сравнительный анализ методов детекции и распознавания, разработаны алгоритмы на основе нейросетевой архитектуры YOLO для детекции полей паспорта, реализован гибридный подход к распознаванию текста.

Результаты работы: разработана информационная система, обеспечивающая автоматическое извлечение персональных данных из изображений паспортов с точностью распознавания более 90%, реализована поддержка пакетной обработки документов, создан удобный графический интерфейс.

Область применения результатов: государственные учреждения, финансовые организации, телекоммуникационные компании и другие организации, работающие с большими объемами персональных документов.

Содержание

Определения, обозначения и сокращения	8
ВВЕДЕНИЕ	10
1 Анализ предметной области	13
1.1 Обзор существующих подходов к обработке документов.....	13
1.1.1 Традиционные методы обработки документов	13
1.1.2 Современные подходы на основе машинного обучения.....	14
1.1.3 Гибридные решения	15
1.2 Особенности работы с электронными копиями паспортов РФ	15
1.2.1 Структура и защитные элементы паспорта	15
1.2.2 Нормативно-правовые аспекты обработки персональных данных ..	16
1.3 Анализ существующих решений и их ограничений	17
1.4 Постановка задачи исследования	20
2 Исследование методов и технологий обработки изображений документов	23
2.1 Классические метода компьютерного зрения.....	23
2.1.1 Предобработка изображений	23
2.1.2 Обнаружение границ и контуров.....	24
2.1.3 Геометрические преобразования	26
2.2 Применение нейронных сетей для детекции объектов	27
2.2.1 Архитектуры сверточных нейронных сетей	27
2.2.2 Семейство моделей YOLO	29
2.2.3 Особенности обучения на малых выборках.....	30
2.3 Технологии оптического распознавания символов	32
2.3.1 Tesseract OCR: возможности и ограничения	32
2.3.2 EasyOCR: современный подход к многоязычному распознаванию .	33
2.3.3 Методы постобработки результатов OCR	35
2.4 Сравнительный анализ методов определения ориентации	36
2.4.1 Методы на основе анализа текста	36
2.4.2 Подходы использования ключевых точек	37
2.4.3 Нейросетевые решения	38

2.4.4 Математическая модель метода на основе confidence score	39
3 Разработка системы обработки паспортов.....	41
3.1 Исследование и анализ подходов к обработке документов	41
3.1.1 Первоначальные эксперименты с классическими методами	41
3.1.2 Исследование применимости нейросетевых моделей для определения ориентации	42
3.1.3 Выбор оптимального технологического стека	43
3.2 Проектирование архитектуры системы	44
3.2.1 Компонентная модель и паттерны проектирования	44
3.2.2 Структуры данных и модель предметной области	45
3.2.3 Динамическое моделирование системы.....	46
3.3 Подготовка данных и обучение моделей машинного обучения.....	50
3.3.1 Формирование обучающей выборки на основе MIDV-2020	50
3.3.2 Разметка данных в Roboflow и подготовка для YOLO	51
3.3.3 Процесс обучения моделей YOLOv8	54
3.3.4 Дообучение OCR для распознавания текстовых полей	54
3.3.5 Анализ результатов обучения и метрики качества.....	55
3.4 Программная реализация модулей обработки	58
3.4.1 Реализация основного процессора обработки документов.....	58
3.4.2 Алгоритм извлечения и выравнивания страницы документа.....	60
3.4.3 Реализация алгоритма определения ориентации	61
3.4.4 Гибридная система оптического распознавания символов.....	62
3.4.5 Реализация интеллектуального fallback-механизма	64
3.4.6 Реализация механизмов валидации и коррекции результатов.....	66
3.4.7 Разработка графического интерфейса пользователя	67
3.5 Экспериментальное исследование и тестирование системы	72
3.5.1 Методика проведения эксперимента.....	72
3.5.2 Анализ производительности системы	73
3.5.3 Оценка точности распознавания полей документа	74
3.5.4 Исследование влияния качества изображений на результаты	76
3.5.5 Анализ типичных ошибок и пути их устранения	78
3.6 Оценка эффективности разработанной системы	80

3.6.1 Сравнение с требованиями технического задания	80
3.6.2 Анализ достигнутых показателей эффективности	82
3.6.3 Рекомендации по практическому применению	83
ЗАКЛЮЧЕНИЕ	85
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	87
ПРИЛОЖЕНИЕ А	91

Определения, обозначения и сокращения

В настоящей выпускной квалификационной работе применяют следующие термины с соответствующими определениями:

Компьютерное зрение – междисциплинарная область, изучающая методы автоматического извлечения, анализа и понимания полезной информации из цифровых изображений.

Обнаружение объектов – задача компьютерного зрения, заключающаяся в обнаружении и локализации объектов определенных классов на изображении.

Оптическое распознавание символов – технология автоматического распознавания текста на изображениях и его преобразования в машиночитаемый формат.

Сверточная нейронная сеть – класс глубоких нейронных сетей, специально разработанных для анализа визуальных изображений.

Bounding box – прямоугольная область на изображении, ограничивающая объект интереса.

Confidence score – численная оценка уверенности модели в правильности предсказания.

Аугментация – способ искусственно увеличить объём данных для обучения модели, применяя к существующим данным разнообразные трансформации, сохраняя их смысл.

Конвейер обработки (pipeline) – последовательность взаимосвязанных этапов обработки данных, где результат выполнения предыдущего этапа служит входными данными для следующего.

В работе используются следующие сокращения:

ИС – информационная система

OCR - Optical Character Recognition (оптическое распознавание символов)

CNN - Convolutional Neural Network (сверточная нейронная сеть)

YOLO - You Only Look Once (архитектура нейронной сети)

API - Application Programming Interface (программный интерфейс приложения)

GPU - Graphics Processing Unit (графический процессор)

mAP - mean Average Precision (средняя точность)

IoU - Intersection over Union (пересечение над объединением)

CLAHE - Contrast Limited Adaptive Histogram Equalization

SIFT - Scale-Invariant Feature Transform

RANSAC - Random Sample Consensus

DPI - Dots Per Inch (точек на дюйм)

FTP - File Transfer Protocol

ВВЕДЕНИЕ

Современное развитие информационных технологий характеризуется стремительным переходом к цифровизации всех сфер деятельности человека. Особую актуальность приобретает автоматизация процессов обработки документов, удостоверяющих личность, поскольку такие документы являются основой для предоставления государственных и коммерческих услуг. Ежедневно миллионы организаций сталкиваются с необходимостью обработки паспортных данных при оформлении договоров, открытии счетов, регистрации пользователей и выполнении других операций.

Традиционный подход к обработке документов, основанный на ручном вводе данных, характеризуется высокими временными затратами, значительной вероятностью ошибок и необходимостью привлечения большого количества персонала. По данным исследований, средняя скорость ручного ввода данных составляет 10-15 документов в час, при этом частота ошибок достигает 3-5% [1]. В условиях возрастающих объемов документооборота такой подход становится неэффективным и требует поиска альтернативных решений.

Развитие технологий компьютерного зрения и машинного обучения открывает новые возможности для автоматизации обработки документов. Современные алгоритмы способны с высокой точностью определять положение документа на изображении, извлекать текстовую информацию и обрабатывать графические элементы. Особый интерес представляют нейросетевые архитектуры семейства YOLO, демонстрирующие высокую скорость и точность при решении задач детекции объектов [2].

Несмотря на значительный прогресс в области автоматической обработки документов, существует ряд нерешенных проблем. К ним относятся: вариативность качества входных изображений, сложность определения правильной ориентации документа, наличие защитных

элементов, затрудняющих распознавание, необходимость обеспечения высокой точности извлечения данных при работе с документами различного качества.

Актуальность темы исследования обусловлена растущей потребностью организаций в эффективных средствах автоматизации документооборота, необходимостью снижения трудозатрат на обработку документов и минимизации ошибок при вводе данных. Разработка специализированной системы для обработки паспортов граждан РФ имеет важное практическое значение для широкого круга организаций.

Объектом исследования является процесс автоматической обработки электронных копий паспортов граждан Российской Федерации.

Предметом исследования выступают методы и алгоритмы компьютерного зрения и машинного обучения, применимые для автоматического извлечения структурированной информации из изображений документов.

Целью работы является разработка информационной системы для автоматизированной обработки потока копий электронных документов с применением современных методов компьютерного зрения и машинного обучения.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ существующих методов и систем обработки изображений документов, выявить их достоинства и недостатки;
2. Исследовать применимость различных подходов к решению задач детекции и распознавания элементов паспорта;
3. Разработать алгоритм определения ориентации документа на изображении с использованием нейросетевых моделей;

4. Создать и обучить модели машинного обучения для детекции основных полей паспорта;
5. Реализовать гибридный подход к распознаванию текстовой информации, обеспечивающий высокую точность в различных условиях;
6. Спроектировать и реализовать программный комплекс с графическим интерфейсом для автоматизированной обработки документов;
7. Провести экспериментальное исследование разработанной системы и оценить ее эффективность.

Методологической основой исследования являются теория обработки изображений, методы машинного обучения, технологии глубокого обучения. В работе использованы методы экспериментального исследования, сравнительного анализа, математического моделирования.

Научная новизна работы заключается в разработке комплексного подхода к обработке паспортов, сочетающего преимущества современных нейросетевых архитектур для детекции объектов с классическими методами распознавания текста, а также в создании итеративного алгоритма определения ориентации документа на основе анализа уверенности детекции.

Практическая значимость работы определяется возможностью применения разработанной системы для автоматизации процессов обработки документов в различных организациях, что позволит существенно сократить временные и трудовые затраты.

1 Анализ предметной области

1.1 Обзор существующих подходов к обработке документов

1.1.1 Традиционные методы обработки документов

Автоматизация обработки документов представляет собой комплексную задачу, решение которой требует применения различных технологий и методов. Исторически первые системы обработки документов базировались на жестко заданных шаблонах и правилах, что ограничивало их применимость узким классом стандартизированных документов [3].

Традиционный подход к обработке документов включает несколько последовательных этапов. Первый этап – получение цифрового изображения документа путем сканирования или фотографирования. Качество полученного изображения существенно влияет на успешность последующей обработки. Второй этап – предварительная обработка изображения, включающая коррекцию яркости и контраста, устранение шумов, выравнивание геометрических искажений. Третий этап – сегментация изображения для выделения областей, содержащих текст, графические элементы и другую значимую информацию. Четвертый этап – распознавание содержимого выделенных областей. Заключительный этап – структурирование и валидация извлеченных данных [3].

Классические методы обработки изображений документов основаны на анализе геометрических и цветовых характеристик. Для определения границ документа применяются алгоритмы детекции краев, такие как оператор Канни или оператор Собеля. Эти алгоритмы анализируют градиенты яркости в изображении и выделяют области с резкими перепадами интенсивности [3]. После обнаружения краев применяется преобразование Хафа для поиска прямых линий, соответствующих границам документа.

Определение ориентации документа в классических подходах основывается на анализе направления текстовых строк. Алгоритмы проекционного профиля строят гистограммы распределения пикселей вдоль

горизонтальной и вертикальной осей, после чего анализируют периодичность пиков, соответствующих текстовым строкам. Однако данный подход показывает низкую эффективность при работе с документами, содержащими сложную структуру или защитные элементы [3].

1.1.2 Современные подходы на основе машинного обучения

Развитие технологий машинного обучения привело к появлению принципиально новых подходов к обработке документов. В отличие от классических методов, основанных на явно заданных правилах, системы машинного обучения способны автоматически выявлять закономерности в данных и адаптироваться к различным типам документов [4].

Сверточные нейронные сети демонстрируют выдающиеся результаты в задачах анализа изображений. Архитектура CNN специально разработана для обработки данных, имеющих сеточную топологию, что делает их идеальными для работы с изображениями документов. Основными компонентами CNN являются сверточные слои, выполняющие операцию свертки входного изображения с обучаемыми фильтрами, слои пулинга для уменьшения пространственных размеров и полносвязные слои для финальной классификации [5].

Особый интерес для задач обработки документов представляют архитектуры, специализированные на детекции объектов. Семейство моделей R-CNN (Region-based CNN) использует двухэтапный подход: сначала генерируются предложения регионов, которые могут содержать объекты интереса, затем каждый регион классифицируется отдельно. Несмотря на высокую точность, такой подход характеризуется значительными вычислительными затратами [6].

Революционным решением стала архитектура YOLO, предложенная в 2016 году. В отличие от двухэтапных детекторов, YOLO рассматривает задачу обнаружения объектов как единую регрессионную проблему,

предсказывая координаты ограничивающих рамок и вероятности классов непосредственно из полного изображения за один проход через сеть. Это обеспечивает высокую скорость работы при сохранении приемлемой точности [6].

1.1.3 Гибридные решения

Практика показывает, что наилучшие результаты в обработке документов достигаются при комбинировании различных подходов. Гибридные решения объединяют преимущества классических алгоритмов компьютерного зрения с возможностями современных нейросетевых архитектур [4].

Типичная архитектура гибридной системы включает несколько уровней обработки. На первом уровне применяются быстрые классические алгоритмы для предварительной фильтрации и нормализации изображений. Это позволяет снизить вычислительную нагрузку на последующие этапы и улучшить качество входных данных для нейросетевых моделей. На втором уровне используются специализированные нейронные сети для решения сложных задач детекции и классификации. Третий уровень отвечает за постобработку результатов и может включать как классические алгоритмы верификации, так и дополнительные нейросетевые модели [7].

Важным преимуществом гибридного подхода является возможность создания отказоустойчивых систем. При невозможности обработки документа одним методом автоматически активируются альтернативные алгоритмы. Это особенно актуально при работе с документами различного качества и происхождения.

1.2 Особенности работы с электронными копиями паспортов РФ

1.2.1 Структура и защитные элементы паспорта

Паспорт гражданина Российской Федерации представляет собой документ установленного образца с четко регламентированной структурой.

Основная информация размещена на второй и третьей страницах документа. Вторая страница содержит фотографию владельца, фамилию, имя, отчество, пол, дату и место рождения. Третья страница включает сведения о выдаче паспорта: дату выдачи, код подразделения и наименование органа, выдавшего документ [8].

Паспорт содержит множество защитных элементов, призванных предотвратить подделку документа. К ним относятся водяные знаки, микротекст, специальная бумага с защитными волокнами, рельефное тиснение, голографические элементы. Эти защитные элементы существенно усложняют автоматическую обработку изображений паспорта, создавая дополнительные визуальные помехи и искажения.

Особую сложность представляет обработка фотографии владельца паспорта. Фотография защищена специальной ламинирующей пленкой с голографическими элементами, которые создают блики и искажения при сканировании или фотографировании. Кроме того, со временем фотография может выцветать или повреждаться, что дополнительно затрудняет ее автоматическое извлечение и обработку.

1.2.2 Нормативно-правовые аспекты обработки персональных данных

Обработка паспортных данных регулируется Федеральным законом "О персональных данных" от 27.07.2006 N 152-ФЗ. Согласно данному закону, паспортные данные относятся к категории персональных данных, обработка которых допускается только с соблюдением установленных требований [9].

Основными принципами обработки персональных данных являются законность и справедливость, ограничение обработки конкретными целями, минимизация объема обрабатываемых данных, точность и актуальность данных, ограничение сроков хранения, обеспечение безопасности. Организации, осуществляющие обработку паспортных данных, обязаны

обеспечить их защиту от неправомерного доступа, уничтожения, изменения, блокирования, копирования и распространения.

При разработке систем автоматической обработки паспортов необходимо предусмотреть технические и организационные меры защиты информации. К техническим мерам относятся шифрование данных, разграничение доступа, аудит действий пользователей, резервное копирование. Организационные меры включают разработку политик безопасности, контроль доступа к помещениям и оборудованию [10].

1.3 Анализ существующих решений и их ограничений

На рынке представлен ряд коммерческих и открытых решений для автоматической обработки документов, удостоверяющих личность. Проведем анализ наиболее значимых систем, применимых для обработки паспортов граждан РФ.

Regula Document Reader SDK представляет собой комплексную платформу для распознавания и верификации документов. Компания Regula Forensics специализируется на разработке программно-аппаратных решений для проверки подлинности документов с 1992 года. Их SDK поддерживает более 13000 шаблонов документов из 251 страны и территории, включая все типы российских паспортов [11]. Технология основана на комбинации классических методов компьютерного зрения и машинного обучения. Решение включает модули проверки подлинности документов через анализ защитных элементов, что особенно важно для организаций финансового сектора. SDK доступен для различных платформ, включая Windows, Linux, macOS, iOS и Android.

Smart ID Engine от российской компании Smart Engines специализируется на мобильном и серверном распознавании документов. Компания была основана в 2013 году как спин-офф Института системного анализа РАН [12]. Технология оптимизирована для работы в режиме

реального времени на устройствах с ограниченными вычислительными ресурсами. Система использует собственные нейросетевые архитектуры, адаптированные под специфику российских документов, включая паспорта старого и нового образца. Особенностью решения является возможность работы без подключения к интернету, что критически важно для мобильных сценариев использования. Smart ID Engine обеспечивает распознавание более 2000 типов документов из 210 юрисдикций.

Beorg beScan. Еще один российский сервис, предлагающий автоматизацию распознавания документов для корпоративного сектора, в частности для банков и страховых компаний [13]. BeScan позиционируется как облачная платформа, доступная через API, что упрощает интеграцию в существующие бизнес-процессы. Особый акцент делается на обработке фотографий и сканов низкого качества, что является частой проблемой при удаленной работе с клиентами. Система использует собственные OCR-технологии для извлечения данных из паспортов и других стандартных документов, позволяя значительно сократить долю ручной проверки.

Открытые решения представлены библиотеками Tesseract OCR и EasyOCR требуют значительной доработки для применения в задачах обработки паспортов, включая разработку модулей детекции полей, определения ориентации и постобработки результатов [14, 15].

Для систематизации сравнения основных характеристик рассмотренных решений они представлены на таблице 1.

Таблица 1 – Анализ аналогов

Характеристика	Regula Document Reader	Smart ID Engine	Beorg beScan	Tesseract + EasyOCR

Ценовая политика	По запросу	По запросу	По запросу	Бесплатно
------------------	------------	------------	------------	-----------

Продолжение таблицы 1

Точность распознавания	до 99%	95-97%	До 99%	85-90%
Поддержка паспортов РФ	Полная	Полная	Полная	Требуется разработки
Проверка подлинности	Да	Ограниченная	Нет	Нет
Работа без интернета	Да	Да	Нет (API)	Да
Извлечение фотографии	Да	Да	Да	Требуется разработки
Открытость алгоритмов	Нет	Нет	Нет	Да

Анализ существующих решений выявил следующие ключевые ограничения:

Коммерческие системы характеризуются закрытой ценовой политикой, требующей индивидуального расчета стоимости для каждого проекта. Это создает барьеры для малых и средних организаций при планировании бюджета. Закрытость алгоритмов не позволяет адаптировать решения под специфические требования организации и создает зависимость от поставщика. Избыточная функциональность коммерческих платформ, ориентированных на обработку множества типов документов, приводит к неоправданному усложнению системы для организаций, работающих исключительно с паспортами РФ.

Открытые решения, несмотря на бесплатность, требуют существенных трудозатрат на разработку полнофункциональной системы. Базовые OCR-библиотеки не включают специализированные модули для работы с паспортами, такие как обнаружение полей, определение ориентации документа, извлечение фотографии и валидация данных. Это превращает создание готовой к промышленному использованию системы на их основе в комплексную и ресурсоемкую инженерную задачу, требующую существенных трудозатрат и высокой квалификации разработчиков.

Таким образом, существует потребность в специализированном решении для обработки паспортов РФ, которое сочетало бы высокую точность распознавания коммерческих систем с доступностью и гибкостью открытых решений. Разработка такой системы на основе современных открытых технологий компьютерного зрения и машинного обучения является актуальной задачей, решение которой позволит существенно снизить барьеры входа для автоматизации документооборота в организациях различного масштаба.

1.4 Постановка задачи исследования

На основании проведенного анализа предметной области можно сформулировать основные требования к разрабатываемой системе обработки паспортов.

Функциональные требования включают автоматическое определение ориентации документа на изображении, извлечение текстовых полей (фамилия, имя, отчество, дата рождения, место рождения), извлечение фотографии владельца с сохранением качества, достаточного для визуальной идентификации, обработку изображений различного качества, включая фотографии, сделанные мобильными устройствами, поддержку пакетной обработки множества документов.

Нефункциональные требования определяют скорость обработки одного документа не более 20 секунд на современном персональном компьютере, работу в автономном режиме без необходимости подключения к интернету, с поддержкой на ОС windows, модульную архитектуру, позволяющую добавлять новую функциональность, соответствие требованиям законодательства по защите персональных данных.

Основной научной задачей является разработка гибридного подхода, сочетающего преимущества современных нейросетевых архитектур для детекции объектов с классическими методами распознавания текста. Такой подход должен обеспечить высокую точность обработки при сохранении приемлемой скорости работы.

Техническими задачами являются исследование применимости архитектуры YOLO для детекции элементов паспорта, разработка алгоритма определения ориентации документа на основе анализа уверенности детекции, создание механизма резервного распознавания для обработки сложных случаев, оптимизация процесса обработки для достижения требуемой производительности.

Ожидаемыми результатами работы являются алгоритмы обработки изображений паспортов, обеспечивающие высокую точность извлечения данных, программная реализация системы с графическим интерфейсом пользователя, экспериментальная оценка эффективности предложенных решений, рекомендации по практическому применению разработанной системы.

Научная новизна исследования заключается в разработке итеративного алгоритма определения ориентации документа на основе анализа confidence score нейросетевой модели, создании гибридного подхода к распознаванию, автоматически выбирающего оптимальный метод для каждого поля документа, применении сегментации на основе полигонов для точного выделения страницы паспорта.

Практическая значимость работы определяется возможностью использования разработанной системы для автоматизации процессов обработки документов в различных организациях без необходимости приобретения дорогостоящих коммерческих лицензий.

2 Исследование методов и технологий обработки изображений документов

2.1 Классические метода компьютерного зрения

2.1.1 Предобработка изображений

Предобработка изображений является ключевым этапом в системах автоматической обработки документов. От качества предобработки напрямую зависит успешность всех последующих этапов анализа. Основной целью предобработки является приведение изображения к виду, оптимальному для дальнейшего анализа, путем устранения различных искажений и помех [3].

Первым шагом предобработки является преобразование цветного изображения в градации серого. Для документов, содержащих преимущественно текстовую информацию, цветовая составляющая часто не несет существенной информации, но значительно увеличивает вычислительную сложность обработки. Преобразование выполняется по формуле (1):

$$Gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

где R, G, B - значения красной, зеленой и синей компонент соответственно. Данные коэффициенты учитывают различную чувствительность человеческого глаза к разным цветам [3].

Следующим важным этапом является нормализация яркости и контраста изображения. Неравномерное освещение при съемке документа приводит к появлению областей с различной яркостью, что затрудняет последующую бинаризацию и распознавание текста. Для решения этой проблемы применяется адаптивная нормализация гистограммы (CLAHE). Алгоритм разделяет изображение на небольшие блоки и выполняет выравнивание гистограммы для каждого блока независимо, что позволяет компенсировать локальные вариации освещенности [18].

Удаление шумов является обязательным этапом предобработки, особенно для изображений, полученных с помощью мобильных камер. Наиболее эффективными для обработки документов являются медианный фильтр и билатеральный фильтр. Медианный фильтр эффективно удаляет импульсные шумы, сохраняя при этом границы объектов. Билатеральный фильтр учитывает как пространственную близость пикселей, так и их яркостное сходство, что позволяет сглаживать шумы, не размывая края текстовых символов [3].

Бинаризация изображения преобразует изображение в градациях серого в двоичное черно-белое изображение. Для документов с неравномерным освещением простая пороговая бинаризация дает неудовлетворительные результаты. Адаптивная бинаризация, такая как метод Оцу или метод Саувола, вычисляет локальный порог для каждого пикселя на основе статистических характеристик его окрестности. Метод Саувола, особенно эффективный для документов с текстом, использует пороговое значение, которое вычисляется по формуле (2):

$$T(x, y) = m(x, y) \times \left[1 + k \times \left(\frac{s(x, y)}{R} - 1 \right) \right] \quad (2)$$

где $T(x, y)$ - пороговое значение для пикселя (x, y) , $m(x, y)$ и $s(x, y)$ - локальное среднее и стандартное отклонение, k - параметр чувствительности (обычно 0.5), R - динамический диапазон стандартного отклонения [3].

2.1.2 Обнаружение границ и контуров

Обнаружение границ документа на изображении является фундаментальной задачей, решение которой позволяет выделить область интереса и устранить фоновые элементы. Классические методы детекции границ основаны на анализе градиентов интенсивности изображения [3].

Оператор Собеля вычисляет приближенное значение градиента яркости изображения. Для каждой точки вычисляются частные производные по

горизонтали (G_x) и вертикали (G_y) с использованием сверточных масок (3) и (4):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

На основе этих значений вычисляются модуль (величина) и направление градиента по формулам (5) и (6) соответственно.

$$G = \sqrt{(G_x^2 + G_y^2)} \quad (5)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (6)$$

Оператор Собеля устойчив к шумам благодаря усреднению в перпендикулярном направлении [3].

Детектор границ Канни является более сложным алгоритмом, обеспечивающим оптимальное обнаружение границ. Алгоритм включает следующие этапы: сглаживание изображения гауссовым фильтром, вычисление градиентов с помощью оператора Собеля, подавление немаксимумов для утончения границ, применение двойной пороговой фильтрации, трассировка границ с использованием гистерезиса. Детектор Канни обеспечивает хорошую локализацию границ и низкий уровень ложных срабатываний [3].

После обнаружения границ применяется алгоритм поиска контуров. В OpenCV реализован алгоритм Сузуки, который строит иерархию контуров, учитывая их вложенность. Для документов наибольший внешний контур

обычно соответствует границам страницы. Контур аппроксимируется многоугольником с помощью алгоритма Дугласа-Пекера, что позволяет получить четыре угловые точки документа [18].

Преобразование Хафа используется для обнаружения прямых линий на изображении. В его основе лежит параметрическое представление прямой в полярных координатах, которое описывается уравнением (7):

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (7)$$

где ρ - расстояние от начала координат до прямой, θ - угол между перпендикуляром к прямой и осью X. Алгоритм строит аккумуляторное пространство в координатах (ρ, θ) и находит локальные максимумы, соответствующие наиболее выраженным прямым линиям. Для документов это позволяет точно определить границы даже при частичном перекрытии или повреждении краев [3].

2.1.3 Геометрические преобразования

Геометрические преобразования необходимы для коррекции перспективных искажений и приведения документа к стандартной ориентации. Основными типами преобразований являются аффинные и проективные преобразования [3].

Аффинное преобразование сохраняет параллельность прямых и отношения расстояний вдоль параллельных прямых. Оно задается матрицей 2×3 и описывается уравнением (8):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & tx \\ c & d & ty \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (8)$$

где (x, y) - исходные координаты, (x', y') - преобразованные координаты, a, b, c, d - параметры масштабирования и поворота, tx, ty - параметры переноса. Аффинное преобразование используется для коррекции небольших искажений и поворота документа [18].

Проектвиное преобразование является более общим и позволяет корректировать перспективные искажения, возникающие при фотографировании документа под углом. Оно задается матрицей 3×3 и описывается уравнением (9):

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (9)$$

где финальные координаты получаются как $x' = x/w'$, $y' = y/w'$. Для вычисления матрицы преобразования необходимо знать соответствие между четырьмя точками исходного изображения и четырьмя точками целевого изображения [3].

Интерполяция при геометрических преобразованиях определяет качество результирующего изображения. Для документов рекомендуется использовать бикубическую интерполяцию, которая обеспечивает хороший баланс между качеством и вычислительной сложностью. Бикубическая интерполяция учитывает 16 соседних пикселей и использует кубические полиномы для вычисления значения интерполированного пикселя [3].

Определение углов документа для перспективного преобразования может выполняться автоматически с использованием алгоритмов детекции углов, таких как детектор Харриса или FAST. Однако для документов более надежным является подход, основанный на пересечении обнаруженных линий границ. После нахождения четырех угловых точек вычисляется матрица перспективного преобразования, приводящая документ к прямоугольному виду стандартного размера [3].

2.2 Применение нейронных сетей для детекции объектов

2.2.1 Архитектуры сверточных нейронных сетей

Сверточные нейронные сети (Convolutional Neural Networks, CNN) революционизировали область компьютерного зрения, обеспечив возможность автоматического извлечения иерархических признаков из

изображений. Архитектура CNN специально разработана для эффективной обработки данных с сеточной структурой, что делает их идеальными для анализа изображений [18].

Основным строительным блоком CNN является сверточный слой, выполняющий операцию свертки входного изображения с набором обучаемых фильтров (ядер). Каждый фильтр извлекает определенный тип признаков, такие как края, углы или текстуры. Математически операция свертки определена на формуле (10):

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \times K(m, n) \quad (10)$$

где I - входное изображение, K - ядро свертки, (i, j) - позиция в выходной карте признаков. Использование общих весов (параметров фильтра) для всего изображения обеспечивает инвариантность к сдвигу и значительно сокращает количество параметров по сравнению с полносвязными сетями [5].

Слои пулинга используются для постепенного уменьшения пространственных размеров карт признаков, что снижает вычислительную нагрузку и обеспечивает инвариантность к небольшим деформациям. Наиболее распространенным является max pooling, выбирающий максимальное значение в каждом окне. Average pooling вычисляет среднее значение, что может быть полезно для сохранения информации о яркости областей [6].

Функции активации вводят нелинейность в сеть, позволяя моделировать сложные зависимости. ReLU (Rectified Linear Unit) является стандартной функцией активации благодаря простоте вычисления и отсутствию проблемы исчезающего градиента. Она вычисляется по формуле (11):

$$ReLU(x) = \max(0, x) \quad (11)$$

Более современные варианты, такие как Leaky ReLU и ELU, решают проблему "мертвых" нейронов, позволяя небольшому градиенту проходить для отрицательных значений [5].

Архитектура ResNet (Residual Network) решает проблему деградации глубоких сетей путем введения остаточных соединений (skip connections). Вместо того чтобы напрямую изучать требуемое отображение $H(x)$, слои сети обучаются аппроксимировать остаточную функцию $F(x)$, которая описывается выражением (12):

$$F(x) = H(x) - x \quad (12)$$

Это позволяет градиентам эффективно распространяться через очень глубокие сети, что критически важно для извлечения высокоуровневых признаков документов [7].

2.2.2 Семейство моделей YOLO

YOLO (You Only Look Once) представляет собой семейство одноэтапных детекторов объектов, которые рассматривают задачу обнаружения как единую регрессионную проблему. В отличие от двухэтапных методов, таких как R-CNN, YOLO обрабатывает изображение целиком за один проход через сеть, что обеспечивает высокую скорость работы [6].

Основная идея YOLO заключается в разделении входного изображения на сетку $S \times S$ ячеек. Каждая ячейка отвечает за предсказание B bounding boxes и соответствующих им confidence scores, а также C условных вероятностей классов. Каждая ограничивающая рамка состоит из 5 предсказаний: x , y , w , h и confidence. Координаты (x, y) представляют центр рамки относительно границ ячейки, w и h - ширину и высоту относительно всего изображения [2].

YOLOv8, последняя версия на момент исследования, включает множество архитектурных улучшений. Backbone сеть основана на CSPDarknet с модификациями для улучшения градиентного потока. Используется Path Aggregation Network (PAN) для эффективной передачи признаков между различными уровнями. Anchor-free подход устраняет необходимость в предопределенных якорных рамках, что упрощает архитектуру и улучшает обобщающую способность [2].

Функция потерь в YOLOv8 состоит из трех компонентов и описывается выражением (13):

$$Loss = \lambda_{coord} \times L_{box} + \lambda_{obj} \times L_{obj} + \lambda_{cls} \times L_{cls} \quad (13)$$

где L_{box} - потери локализации (используется CIOU loss), L_{obj} - потери объектности (binary cross-entropy), L_{cls} - потери классификации (cross-entropy), λ - весовые коэффициенты для балансировки компонентов [2].

Для задачи обнаружения элементов паспорта YOLO обладает рядом преимуществ: высокая скорость обработки (до 155 FPS на GPU для YOLOv8n), хорошая точность даже на небольших объектах благодаря архитектуре сетей пирамидальных признаков (Feature Pyramid Network, FPN), простота обучения на собственных наборах данных, а также возможность обнаружения множественных объектов различных классов одновременно. Эти характеристики делают YOLO идеальным выбором для обработки документов в режиме реального времени [20].

2.2.3 Особенности обучения на малых выборках

Обучение моделей глубокого обучения для специфических задач, таких как обработка паспортов, часто сталкивается с проблемой ограниченного количества обучающих данных. Сбор и разметка большого датасета паспортов затруднены по причинам конфиденциальности и трудоемкости процесса аннотирования [18].

Transfer learning (перенос обучения) является одной из важнейших техник для работы с малыми выборками. Вместо обучения модели с нуля используется предобученная на большом датасете модель (например, COCO dataset), веса которой дообучаются на целевой задаче. Это позволяет использовать общие признаки, извлеченные из миллионов изображений, адаптируя только высокоуровневые слои под специфику паспортов [5].

Аугментация данных имеет существенное значение для увеличения эффективного размера обучающей выборки. Для документов применяются следующие виды аугментаций:

- Геометрические: повороты ($\pm 15^\circ$), масштабирование (0.8–1.2), перспективные искажения;
- Фотометрические: изменение яркости ($\pm 25\%$), контраста ($\pm 20\%$), добавление шума;
- Специфические для документов: имитация бликов, размытие, искусственные тени.

Важно, чтобы аугментации отражали реальные условия съемки документов [21].

Стратегии обучения для малых выборок включают использование меньшего learning rate (0.001 вместо стандартного 0.01) для предотвращения переобучения, применение сильной регуляризации (dropout 0.5, weight decay 0.0005), раннюю остановку обучения при отсутствии улучшения на валидационной выборке, использование ансамблей моделей для повышения робастности [22].

Метрики оценки должны учитывать дисбаланс классов, характерный для детекции полей документа. Помимо стандартных precision и recall, используются:

- mAP – усредненная точность по всем классам и порогам IoU;

- F1 – score для каждого класса отдельно;
- Confusion matrix для анализа типичных ошибок.

Именно валидация на отложенной выборке с документами различного качества **позволяет объективно оценить** обобщающую способность модели [4].

2.3 Технологии оптического распознавания символов

2.3.1 Tesseract OCR: возможности и ограничения

Tesseract представляет собой наиболее распространенную открытую систему оптического распознавания символов, изначально разработанную Hewlett-Packard в 1985-1994 годах и переданную в open source Google в 2005 году. Современная версия Tesseract 4.x использует нейросетевую архитектуру LSTM (Long Short-Term Memory) для распознавания строк текста [14].

Архитектура Tesseract включает несколько базовых компонентов. Модуль анализа страницы определяет области текста и их ориентацию, используя алгоритмы морфологической обработки и анализа связных компонент. Детектор строк текста применяет алгоритм базовых линий для определения границ текстовых строк. LSTM-сеть обрабатывает изображение строки как последовательность, предсказывая символы слева направо. Языковая модель использует словари и n-граммы для коррекции результатов распознавания [14].

Процесс распознавания в Tesseract начинается с бинаризации изображения с использованием адаптивного порога. Затем выполняется анализ связных компонент для выделения потенциальных символов. Компоненты группируются в слова и строки на основе пространственной близости и выравнивания. LSTM-сеть обрабатывает каждую строку, генерируя последовательность символов с оценками уверенности. Постпроцессор применяет языковую модель для исправления ошибок [15].

Tesseract демонстрирует высокую точность на качественных сканах с четким черным текстом на белом фоне. Поддержка более 100 языков и возможность обучения на пользовательских шрифтах делают его универсальным решением. Однако существуют значительные ограничения: чувствительность к качеству изображения и предобработке, сложности с распознаванием текста на сложном фоне, проблемы с нестандартными шрифтами и рукописным текстом, отсутствие встроенной детекции и выравнивания текста [18].

Для оптимизации работы Tesseract с паспортами применяются следующие настройки:

- Page Segmentation Mode (PSM) 6 или 7 для отдельных текстовых полей;
- Ограничение набора символов через whitelist для полей с известным форматом;
- Использование русскоязычной модели 'rus' с дополнительным обучением на примерах паспортных шрифтов;
- Тонкая настройка параметров бинаризации для документов с защитными элементами.

2.3.2 EasyOCR: современный подход к многоязычному распознаванию

EasyOCR представляет собой современную библиотеку оптического распознавания символов, основанную на глубоком обучении. В отличие от Tesseract, EasyOCR изначально разработана с использованием нейросетевых технологий и оптимизирована для работы с изображениями различного качества [15].

Архитектура EasyOCR состоит из трех основных модулей. Детектор текста на основе CRAFT (Character Region Awareness for Text Detection) обнаруживает области текста произвольной формы и ориентации. Модуль

распознавания использует CRNN (Convolutional Recurrent Neural Network) с механизмом внимания для преобразования изображений текста в последовательности символов. Декодер применяет CTC (Connectionist Temporal Classification) или attention-based декодирование для получения финальной текстовой строки [15].

CRAFT детектор работает на уровне символов, предсказывая тепловые карты для центров символов и связей между ними. Это позволяет точно определять границы слов даже для изогнутого или повернутого текста. Алгоритм постобработки объединяет символьные регионы в слова и строки на основе предсказанных связей [4].

CRNN архитектура для распознавания состоит из трех частей:

- Сверточные слои извлекают визуальные признаки из изображения текста;
- Рекуррентные слои (Bidirectional LSTM) моделируют последовательную природу текста;
- Transcription слой преобразует последовательность признаков в текст.

Использование внимания позволяет модели фокусироваться на релевантных частях изображения при предсказании каждого символа [15].

Преимущества EasyOCR для обработки паспортов включают устойчивость к различным ориентациям и искажениям текста, хорошую работу на изображениях низкого качества, поддержку 80+ языков с возможностью многоязычного распознавания, отсутствие необходимости в сложной предобработке. Недостатками являются более высокие вычислительные требования по сравнению с Tesseract, меньшая скорость работы, особенно на CPU, сложность тонкой настройки для специфических задач [15].

2.3.3 Методы постобработки результатов OCR

Результаты оптического распознавания символов часто содержат ошибки, обусловленные качеством изображения, особенностями шрифта или алгоритмическими ограничениями OCR-движков. Постобработка результатов является важным этапом, позволяющим существенно повысить итоговую точность распознавания [14].

Коррекция на основе словарей представляет классический подход к улучшению результатов OCR. Для каждого распознанного слова вычисляется расстояние редактирования (расстояние Левенштейна) до слов из эталонного словаря. Если минимальное расстояние не превышает заданный порог, слово заменяется на ближайшее из словаря. Данный метод эффективен для полей с ограниченным набором возможных значений, таких как имена или названия городов. Однако применение общих словарей может приводить к некорректным исправлениям для редких или составных слов.

Контекстная коррекция учитывает семантику и структуру документа. Для паспортов известно, что поле отчества должно заканчиваться на определенные суффиксы (-ович, -евич, -ич для мужчин, -овна, -евна, -ична для женщин). Если распознанный текст не соответствует этому правилу, применяются эвристические алгоритмы коррекции. Аналогично, дата рождения должна соответствовать формату ДД.ММ.ГГГГ и находиться в разумном диапазоне лет [10].

Статистические языковые модели позволяют оценить вероятность последовательности символов или слов. N-граммные модели, обученные на корпусе текстов соответствующей тематики, используются для ранжирования альтернативных вариантов распознавания. Современные подходы применяют нейросетевые языковые модели на основе архитектур Transformer, которые учитывают долгосрочные зависимости в тексте [19].

Ансамблевые методы комбинируют результаты нескольких OCR-движков для повышения надежности. Простейший подход - голосование большинством, когда выбирается вариант, предложенный большинством движков. Более сложные методы учитывают confidence scores каждого движка и исторические данные о их точности для разных типов текста. Взвешенное голосование позволяет адаптивно выбирать наиболее надежный результат [4].

Специфичные для документов правила постобработки учитывают структурные особенности. Для паспортов это включает проверку контрольных сумм номера документа, валидацию кода подразделения по установленному формату, сопоставление места рождения с базой населенных пунктов. Нарушение таких правил является индикатором ошибки распознавания и запускает процедуру коррекции [8].

2.4 Сравнительный анализ методов определения ориентации

2.4.1 Методы на основе анализа текста

Определение правильной ориентации документа является фундаментальной задачей в системах автоматической обработки [20]. Методы, основанные на анализе текста, используют особенности расположения и структуры текстовых элементов для определения корректной ориентации [16].

Метод проекционных профилей строит гистограммы распределения черных пикселей вдоль горизонтальной и вертикальной осей. Для правильно ориентированного текста горизонтальный профиль показывает четкие пики, соответствующие текстовым строкам, с регулярными промежутками между ними. Анализ автокорреляционной функции профиля позволяет определить периодичность и, следовательно, правильную ориентацию. Однако данный метод показывает низкую эффективность для документов со сложной структурой или многоколоночным текстом [3].

Анализ направления штрихов основан на том факте, что в большинстве алфавитов вертикальные штрихи преобладают над горизонтальными. Алгоритм вычисляет градиенты изображения и строит гистограмму направлений. Для правильно ориентированного текста пик гистограммы находится около 90° (вертикальное направление). Метод устойчив к небольшим поворотам, но может давать неверные результаты для текстов с преобладанием курсива или декоративных шрифтов [3].

Метод анализа ascenders и descenders использует статистические свойства латинского и кириллического алфавитов. Ascenders (верхние выносные элементы) встречаются чаще, чем descenders (нижние выносные элементы). Алгоритм определяет базовую линию текста и подсчитывает количество пикселей выше и ниже нее. Правильная ориентация соответствует максимальной асимметрии в пользу верхних элементов. Метод эффективен для печатного текста, но требует точного определения базовых линий [18].

2.4.2 Подходы использования ключевых точек

Методы на основе ключевых точек извлекают и анализируют характерные особенности изображения, инвариантные к определенным преобразованиям. Эти подходы показывают хорошую устойчивость к шумам и искажениям [21].

SIFT (Scale-Invariant Feature Transform) детектор находит ключевые точки, инвариантные к масштабу и повороту. Для определения ориентации документа сравниваются дескрипторы данных точек тестового изображения с эталонными дескрипторами правильно ориентированного документа. Геометрическая верификация с использованием RANSAC позволяет определить преобразование между изображениями и, следовательно, угол поворота. Недостатком является необходимость в эталонном изображении и высокая вычислительная сложность [21].

ORB (Oriented FAST and Rotated BRIEF) представляет собой эффективную альтернативу SIFT. Детектор FAST находит углы на изображении, а дескриптор BRIEF кодирует окрестности ключевых точек. ORB добавляет вычисление ориентации для каждой из этих точек, что позволяет определить доминирующее направление в изображении. Статистический анализ ориентаций ключевых точек может указать на общий поворот документа. Метод работает значительно быстрее SIFT, но менее устойчив к сильным искажениям [22].

Метод доминирующих направлений анализирует распределение направлений градиентов в областях с высокой текстурой. Для документов такими областями являются текстовые блоки. Алгоритм строит взвешенную гистограмму направлений, где вес определяется величиной градиента. Доминирующие пики в гистограмме соответствуют основным направлениям в документе (обычно 0° , 90° , 180° , 270°). Метод прост в реализации и показывает хорошие результаты для документов с четкой структурой.

2.4.3 Нейросетевые решения

Современные подходы к определению ориентации документов все чаще используют глубокое обучение, демонстрируя превосходные результаты в сложных случаях.

Классификационный подход рассматривает задачу определения ориентации как классификацию на 4 класса (0° , 90° , 180° , 270°). CNN архитектуры, такие как ResNet или EfficientNet, обучаются на размеченных данных документов в различных ориентациях. Преимуществом является высокая точность ($>99\%$) и устойчивость к различным типам искажений. Недостатком является необходимость в большом объеме обучающих данных и ограничение дискретными углами поворота [18].

Регрессионный подход предсказывает непрерывный угол поворота, что позволяет обрабатывать документы с произвольной ориентацией.

Архитектура включает CNN backbone для извлечения признаков и регрессионную голову для предсказания угла. При этом функция потерь должна учитывать цикличность углов, что отражено в выражении (14):

$$L = \min(|\theta_{\text{pred}} - \theta_{\text{true}}|, 360^\circ - |\theta_{\text{pred}} - \theta_{\text{true}}|) \quad (14)$$

Этот подход более гибкий, но требует тщательной настройки и может быть менее стабильным [17].

Инновационный подход, предложенный в данной работе, использует confidence scores детектора объектов YOLO для определения ориентации. В отличие от классических методов, требующих специального обучения модели для задачи определения ориентации, данный подход опирается на наблюдение, что модель детекции, обученная на правильно ориентированных документах, демонстрирует максимальную уверенность при обработке изображений в канонической ориентации.

Сравнительный анализ методов показывает, что выбор оптимального подхода зависит от специфики задачи. Для высококачественных сканов по основным 4 направлениям документа эффективны классические методы анализа текста. Для фотографий документов в различных условиях нейросетевые подходы демонстрируют лучшие результаты. Предложенный метод на основе confidence scores обеспечивает хороший баланс между точностью и простотой реализации, что делает его отличным решением для применения.

2.4.4 Математическая модель метода на основе confidence score

Предложенный метод определения ориентации основан на гипотезе, что нейросетевая модель детекции, обученная на правильно ориентированных документах, демонстрирует максимальную уверенность при обработке изображений в канонической ориентации.

Математически задача формулируется как поиск оптимального угла поворота θ^* , который максимизирует функцию оценки, как показано в выражении (15):

$$\theta^* = \arg \max_{\theta \in \{0^\circ, 180^\circ\}} S(\theta) \quad (15)$$

где $S(\theta)$ - функция оценки качества детекции при повороте на угол θ . Она вычисляется как взвешенная сумма оценок уверенности по всем обнаруженным объектам, за исключением лица, согласно формуле (16):

$$S(\theta) = \sum_{i=1}^N w_i \cdot c_i(\theta) \cdot I[class_i \neq face] \quad (16)$$

где N - количество детектированных объектов, $c_i(\theta)$ - confidence score i -го объекта при повороте θ , w_i - весовой коэффициент для класса объекта, I - индикаторная функция, исключающая класс "face" из расчета.

Дополнительно вводится функция детекции лица $F(\theta)$, принимающая значение 1 при успешном обнаружении лица и 0 в противном случае. Итоговая функция оценки $Q(\theta)$, объединяющая оба критерия, представлена в выражении (17):

$$Q(\theta) = \alpha \cdot S(\theta) + \beta \cdot F(\theta) \quad (17)$$

где $\alpha = 0.8$ и $\beta = 0.2$ - эмпирически подобранные весовые коэффициенты.

3 Разработка системы обработки паспортов

3.1 Исследование и анализ подходов к обработке документов

3.1.1 Первоначальные эксперименты с классическими методами

Разработка системы автоматической обработки паспортов началась с исследования традиционных методов компьютерного зрения. Первый прототип системы использовал библиотеку OpenCV для обнаружения лица с помощью каскадных классификаторов Хаара и Tesseract OCR для распознавания текста [18].

Алгоритм первого прототипа включал следующие этапы: обнаружение лица на изображении с использованием предобученного классификатора, определение центра обнаруженного лица через координаты ограничивающего прямоугольника, расчет фиксированных смещений от центра лица для локализации текстовых полей, преобразование изображения в градации серого и повышение контраста, применение OCR к выделенным областям.

Экспериментальная оценка показала низкую эффективность данного подхода. Точность распознавания текста составляла менее 60% даже на высококачественных изображениях. Анализ результатов выявил следующие критические проблемы: невозможность обработки документов с произвольной ориентацией, жесткая привязка координат полей не учитывала вариативность расположения элементов в разных паспортах, захват избыточных областей при извлечении полей снижал качество OCR.

В попытке улучшить результаты были исследованы различные методы предобработки изображений: адаптивная бинаризация для улучшения контраста текста, морфологические операции для устранения шумов и соединения разрывов в символах, медианная фильтрация для подавления импульсных помех. Также тестировалось применение библиотеки Natasha для валидации русских имен, однако это не дало положительных результатов из-за многонационального характера имен в российских паспортах [17].

3.1.2 Исследование применимости нейросетевых моделей для определения ориентации

Критической проблемой при автоматической обработке документов является определение правильной ориентации изображения. В рамках исследования были реализованы и протестированы несколько подходов к решению данной задачи.

Первым был исследован метод определения ориентации по машиночитаемой зоне, или MRZ (Machine-Readable Zone), представляющей собой специальную область с данными в стандартизированном формате для автоматического считывания. Для этого были проанализированы специализированные библиотеки [22]. Однако дальнейший анализ показал, что паспорта старого образца, составляющие значительную долю обрабатываемых документов, не содержат такую зону, что делает данный подход неприменимым для создания универсальной системы.

Следующий эксперимент заключался в реализации итеративного алгоритма определения ориентации. Суть метода состояла в последовательном повороте изображения с шагом 5 градусов в диапазоне от -180° до $+180^\circ$, применении OCR и детекции лица для каждой ориентации, выборе угла с максимальным количеством распознанных элементов. Метод показал крайне низкую производительность – обработка одного изображения занимала много времени, при этом точность определения оставалась неудовлетворительной.

Для решения задачи была предпринята попытка обучения нейросетевой модели регрессии угла поворота. В качестве архитектуры использовалась ResNet18 с модифицированным выходным слоем для предсказания непрерывного значения угла. Подготовка обучающих данных выполнялась на основе датасета MIDV-2020 [23], из JSON-аннотаций которого извлекались координаты углов документа для вычисления истинного угла поворота.

Несмотря на использование современной архитектуры и качественных данных, результаты оказались неудовлетворительными – только 3-5% изображений были корректно ориентированы после применения модели. Анализ показал, что модель не смогла обучиться надежным признакам для определения ориентации документа, что может быть связано со сложностью задачи регрессии угла в широком диапазоне.

Параллельно исследовался классический метод компьютерного зрения SIFT для определения ориентации через сопоставление ключевых точек [21]. Алгоритм включал создание эталонного набора из правильно ориентированных изображений, извлечение SIFT-дескрипторов, поиск соответствий между эталоном и тестовым изображением, вычисление матрицы гомографии методом RANSAC. Подход показал низкую эффективность - менее 10% изображений были повернуты некорректно, основными проблемами стали ложные соответствия из-за повторяющихся паттернов защитной сетки паспорта и искажения при применении найденной гомографии.

3.1.3 Выбор оптимального технологического стека

На основании результатов проведенных экспериментов был сделан вывод о необходимости применения современных нейросетевых архитектур детекции объектов.

Для реализации был выбран YOLOv8, поскольку на нём доступна наибольшая документация и практический опыт [24]. Важным фактором выбора стала также поддержка задачи instance segmentation в YOLOv8-seg.

Для программной реализации использовалась библиотека Ultralytics, предоставляющая высокоуровневый Python API для работы с YOLO [20]. Библиотека обеспечивает простую интеграцию моделей, удобные инструменты для обучения и инференса, автоматическую обработку

различных форматов данных, встроенную поддержку аугментаций и метрик качества.

Для задачи оптического распознавания символов был выбран гибридный подход, комбинирующий два OCR-движка. Tesseract OCR используется как основной движок [14]. EasyOCR применяется как резервный механизм [15]. Комбинирование двух подходов позволяет достичь максимальной надёжности распознавания.

3.2 Проектирование архитектуры системы

3.2.1 Компонентная модель и паттерны проектирования

Архитектура разработанной ИС основана на принципах модульности и слабой связанности компонентов, что обеспечивает гибкость и возможность дальнейшего расширения функциональности. Система построена по трехуровневой архитектуре и представлена на рисунке 1.

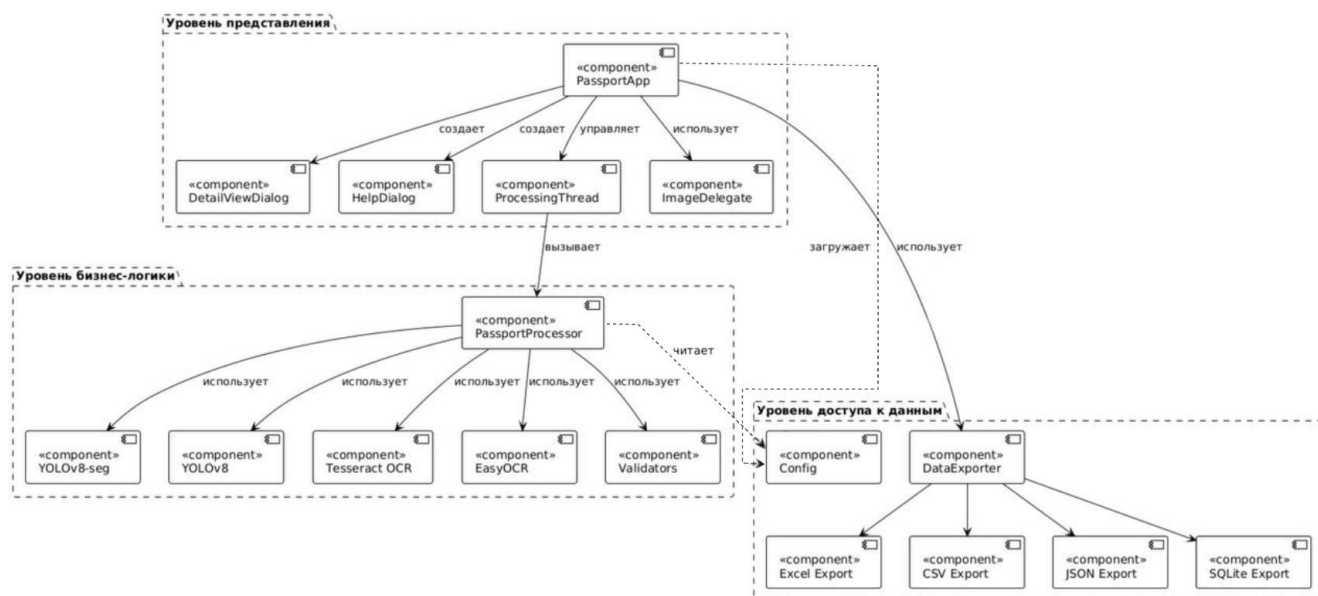


Рисунок 1 - Диаграмма компонентов

Уровень представления реализован с использованием фреймворка PyQt5 и включает главное окно приложения (PassportApp), диалоговые окна для детального просмотра и настройки, а также специализированные виджеты для отображения результатов. Данный уровень отвечает за

взаимодействие с пользователем, визуализацию данных и обработку пользовательских команд.

Уровень бизнес-логики содержит основные компоненты обработки. Он включает в себя центральный модуль PassportProcessor, координирующий весь процесс, модули детекции и сегментации на базе YOLOv8 для локализации элементов паспорта, гибридную систему OCR, объединяющую Tesseract и EasyOCR, а также компоненты для валидации и постобработки результатов.

Уровень доступа к данным обеспечивает экспорт результатов в различные форматы (Excel, CSV, JSON, SQLite), загрузку и сохранение конфигурации системы, а также функционал для сопоставления распознанных данных с информацией из локальных файлов.

Взаимодействие между уровнями осуществляется через четко определенные интерфейсы, что позволяет модифицировать или заменять отдельные компоненты без влияния на работу системы в целом. Внешние библиотеки и предобученные модели инкапсулированы в отдельном слое, что упрощает их обновление и поддержку. Такая архитектура обеспечивает масштабируемость решения и возможность добавления новых типов документов путем интеграции соответствующих моделей обработки без изменения базовой структуры приложения.

3.2.2 Структуры данных и модель предметной области

Для представления извлеченной информации о паспорте разработана структура данных PassportData, реализованная как Python dataclass. Данный подход обеспечивает автоматическую генерацию конструктора, методов сравнения и строкового представления, что упрощает работу с данными. Полное определение структуры представлено в листинге А.1.

Структура включает следующие группы атрибутов: текстовые поля документа (фамилия, имя, отчество, дата и место рождения, номер паспорта),

изображения (исходное изображение, извлеченная страница, фотография владельца), метаданные обработки (оценки уверенности для каждого поля, список ошибок обработки).

Для конфигурирования системы разработан класс `Config`, централизующий все настройки: пути к моделям машинного обучения, параметры обработки изображений, настройки OCR-движков, пороговые значения для валидации. Использование единого класса конфигурации упрощает настройку системы и позволяет легко сохранять и загружать различные профили настроек.

3.2.3 Динамическое моделирование системы

Динамические аспекты работы системы отражены в диаграммах взаимодействия.

Диаграмма последовательности обработки документа, представлена на рисунке 2. Она детализирует процесс обработки изображений от момента

загрузки до сохранения результатов.

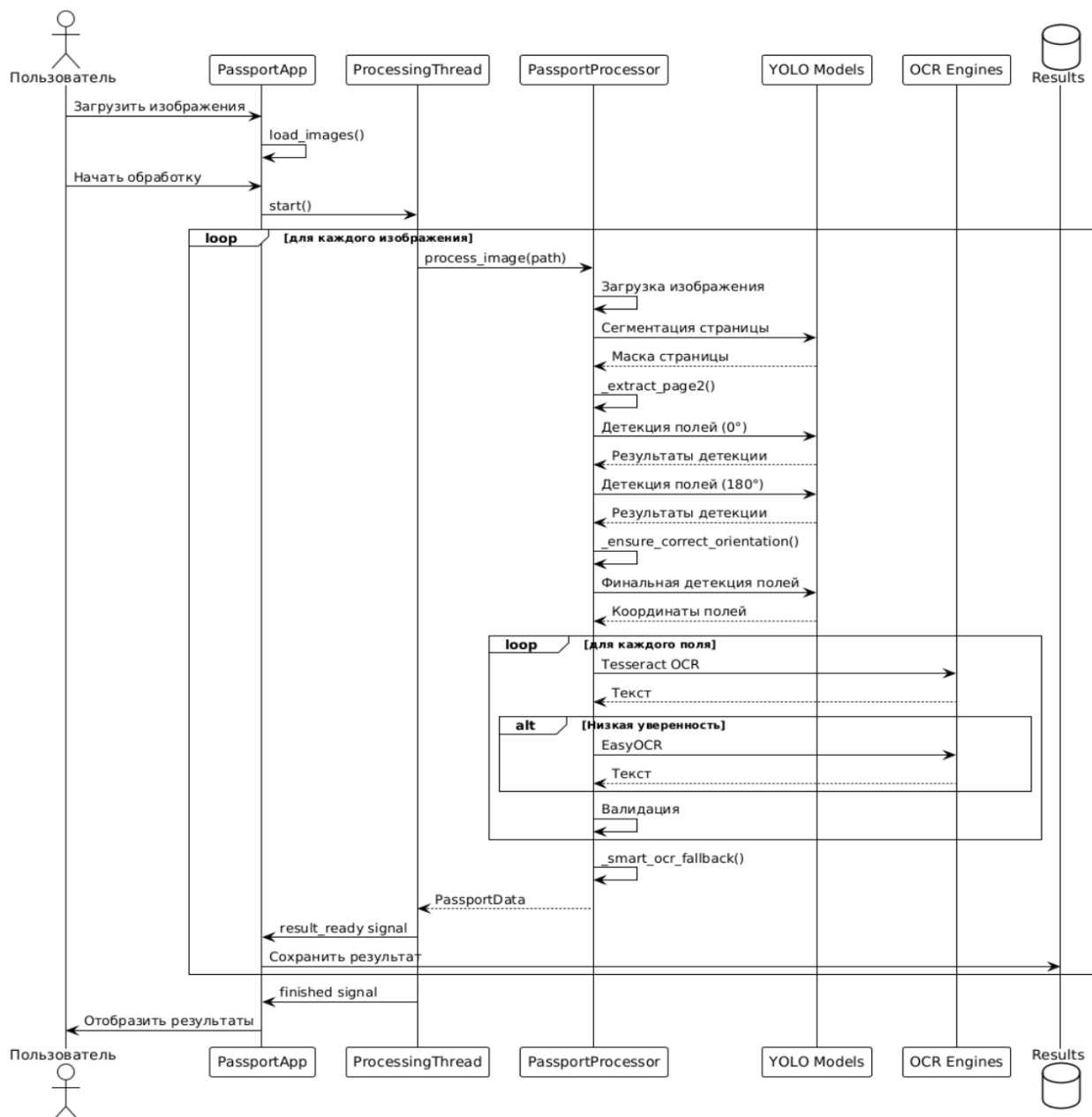


Рисунок 2 - Диаграмма последовательности

Процесс начинается с выбора пользователем изображений и запуска обработки. Главное окно создает фоновый поток ProcessingThread, который последовательно обрабатывает каждое изображение через PassportProcessor. Для каждого изображения выполняется цепочка операций. Сначала выполняется сегментация страницы паспорта моделью YOLOv8-seg, которая выделяет область документа на изображении. Затем происходит определение правильной ориентации документа путем параллельного анализа результатов

детекции при повороте на 0° и 180° . После установления корректной ориентации выполняется детекция всех текстовых полей с помощью модели YOLOv8. Завершающим этапом является применение гибридного OCR-распознавания, где система автоматически выбирает между Tesseract и EasyOCR в зависимости от характеристик конкретного поля.

Поток обработки периодически отправляет сигналы о прогрессе выполнения главному окну, которое обновляет индикатор прогресса и информирует пользователя о текущем состоянии. По завершении обработки каждого документа результаты передаются обратно в главное окно для отображения в табличном виде.

Диаграмма деятельности представлена на рисунке 3. Она представляет алгоритм работы системы в виде потока управления. Она наглядно показывает все развилки и циклы в процессе обработки, включая обработку ошибок и активацию резервных механизмов распознавания.

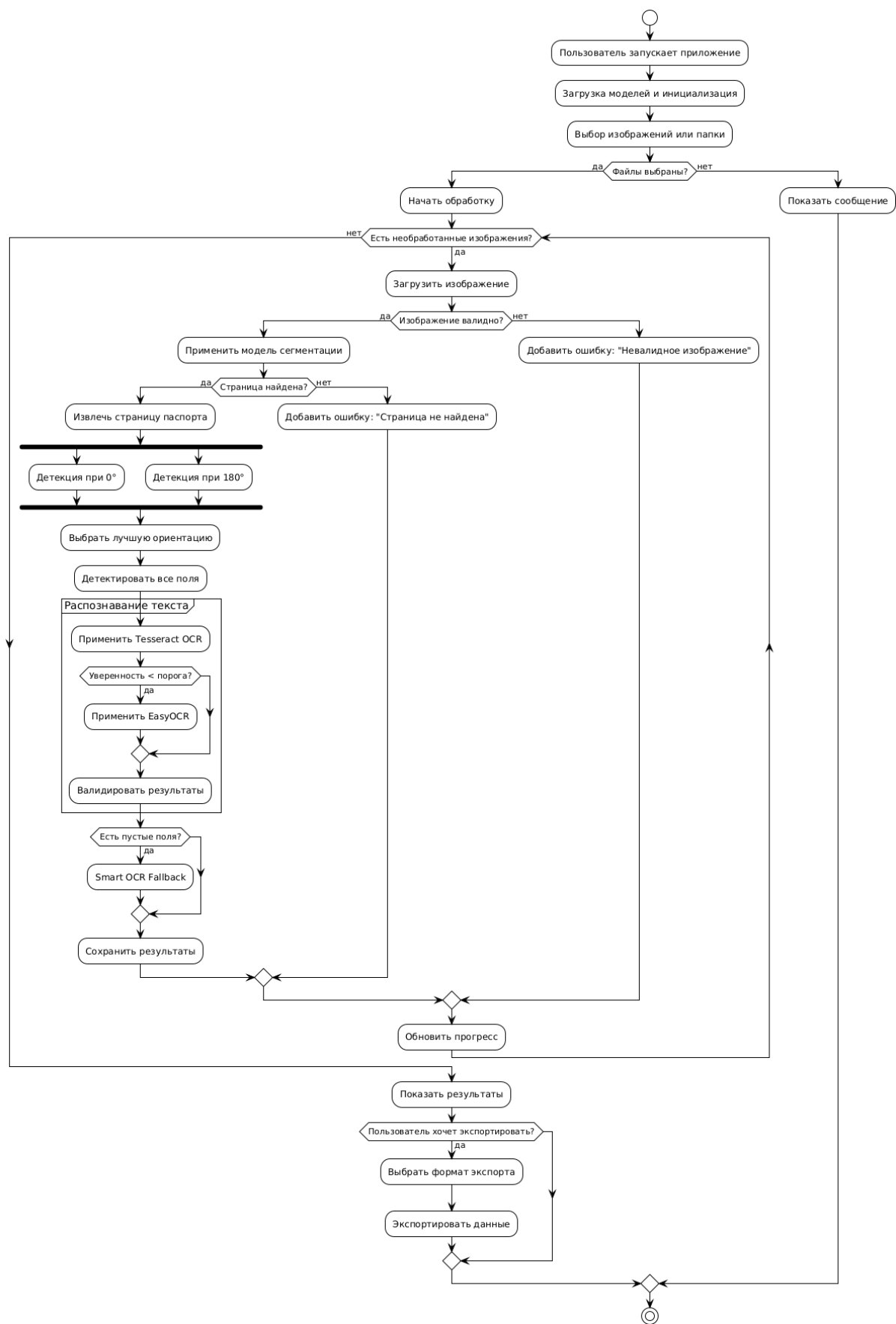


Рисунок 3 - Диаграмма деятельности

Основным элементом алгоритма является блок определения ориентации, который реализован как параллельный процесс. Система одновременно выполняет детекцию полей при двух возможных ориентациях документа. Выбирается вариант с максимальным значением комплексной оценки, учитывающей количество обнаруженных полей, средний показатель уверенности детекции и наличие фотографии владельца.

После основной обработки активируется механизм проверки результатов. Если количество распознанных полей меньше ожидаемого или обнаружены признаки некорректного распознавания, запускается резервный алгоритм обработки. Этот механизм выполняет полное сканирование изображения с помощью дообученной версии EasyOCR и по примерным координатам находит значение для отдельных текстовых полей.

3.3 Подготовка данных и обучение моделей машинного обучения

3.3.1 Формирование обучающей выборки на основе MIDV-2020

Качество моделей машинного обучения напрямую зависит от качества и репрезентативности обучающих данных. В качестве основы для формирования датасета использовался MIDV-2020 - полностью аннотированная база синтетически сгенерированных изображений документов, удостоверяющих личность [23]. С целью повышения робастности и обобщающей способности моделей, исходный датасет был расширен за счет синтетических данных. Была сгенерирована дополнительная выборка фотореалистичных изображений российских паспортов с помощью генеративной нейросетевой модели.

Процесс получения и подготовки данных включал несколько этапов. После заполнения формы на официальном сайте проекта и получения доступа, данные были загружены через FTP-протокол с использованием клиента FileZill. Общий объем датасета составил около 15 ГБ, включая изображения документов 10 стран в различных условиях съемки.

Из полного датасета были извлечены изображения российских паспортов. Каждое изображение сопровождалось JSON-файлом с детальными аннотациями, включающими координаты всех текстовых полей, координаты углов документа, эталонные тексты полей (ground truth).

Анализ извлеченных данных показал высокое качество разметки - координаты полей были точными, тексты соответствовали изображениям. Однако выявилась проблема: координаты в аннотациях ограничивали текст точно по контуру символов, без создания необходимых для качественного распознавания отступов. Поэтому было принято решение использовать MIDV-2020 как основу для обучения моделей обнаружения, которые создают рамки с разумными отступами. В то же время эталонные текстовые данные были использованы для дообучения модели распознавания EasyOCR.

3.3.2 Разметка данных в Roboflow и подготовка для YOLO

Платформа Roboflow была выбрана для разметки данных благодаря удобному веб-интерфейсу, поддержке различных форматов экспорта и широким возможностям ручной аннотации. [18]. При выборе инструмента учитывались рекомендации, приведённые в документации YOLO [20]. Процесс разметки выполнялся отдельно для каждой модели с учетом специфики решаемой задачи.

Для модели сегментации второй страницы паспорта создавался проект типа Instance Segmentation. В проект загружались 128 изображений, включающих как данные из MIDV-2020, так и дополнительно собранные фотографии для увеличения вариативности. Разметка выполнялась путем создания полигона, точно обводящего границы второй страницы паспорта. Использование полигональной разметки вместо прямоугольных bounding box обеспечивает возможность корректной обработки повернутых документов.

После завершения ручной разметки изображения были автоматически разделены на обучающую, валидационную и тестовую выборки, содержащие

соответственно 270, 25 и 13 изображений. На этапе предобработки применялась автоориентация, а также масштабирование до размера 640×640 пикселей с сохранением пропорций и добавлением белых полей. Для повышения устойчивости модели к вариативности условий съёмки использовались следующие аугментации:

- поворот изображений в диапазоне от -10° до $+10^\circ$;
- изменение яркости в пределах $\pm 10\%$;
- изменение экспозиции в пределах $\pm 9\%$;
- размытие до 0,4 пикселя;
- добавление шума до 0,18 % пикселей.

На основе каждого исходного изображения генерировалось три аугментированных варианта, что позволило существенно расширить объем обучающей выборки.

Важным этапом стала конвертация аннотаций из формата COCO, генерируемого Roboflow, в формат YOLOv8-seg. Разработанный конвертер извлекает координаты полигонов из COCO-аннотаций, нормализует их относительно размеров изображения и сохраняет в текстовые файлы в формате YOLO.

Для модели детекции полей документа процесс разметки имел свои особенности. Создавался проект Object Detection с шестью классами: photo (фотография), surname (фамилия), name (имя), patronymic (отчество), birth_date (дата рождения), birth_place (место рождения). Размечались 181 изображений второй страницы паспорта, для каждого поля создавался bounding box с отступами 5-10 пикселей для улучшения качества последующего OCR.

При выборе аугментаций учитывалась специфика задачи - повороты не применялись, чтобы сохранить правильную ориентацию текста относительно bounding box, использовались следующие параметры:

- поворот от -5° до $+5^\circ$;
- преобразование в градации серого (на 15% изображений);
- изменение яркости в пределах $\pm 25\%$;
- размытие до 1 пикселя;
- шум до 1.37% пикселей.

После применения аугментаций получено 453 изображения для обучения.

Третья модель использовалась для точного обнаружения строки с номером паспорта. Данный проект был реализован в формате Object Detection с одним классом `passport_number`. В него вошло 94 исходных изображения второй страницы, на которых вручную размечалась строка с номером.

Благодаря аугментациям обучающая выборка увеличилась до 270 изображений, валидационная — до 25, тестовая — до 13.

Применялись следующие аугментации:

- поворот от -10° до $+10^\circ$;
- яркость: $\pm 10\%$;
- экспозиция: $\pm 9\%$;
- размытие до 0.4 пикселя;
- шум до 0.18% пикселей.

Каждое исходное изображение также дополнялось тремя аугментированными, что обеспечивало устойчивость модели к поворотам и освещению.

3.3.3 Процесс обучения моделей YOLOv8

Обучение моделей выполнялось на рабочей станции с GPU NVIDIA RTX 4070 Laptop, что обеспечило приемлемую скорость обучения - около 25 минут за 100 эпох для модели обнаружения. Для каждой задачи выбиралась оптимальная конфигурация модели из линейки YOLOv8 [20].

Для сегментации страницы паспорта использовалась модель YOLOv8m-seg (medium). Выбор обусловлен балансом между точностью и скоростью - модель достаточно большая для надежной сегментации, но не избыточна для задачи с одним классом. Параметры обучения: 100 эпох, размер изображения 640×640, batch size 16, оптимизатор AdamW с начальным learning rate 0.001.

Для детекции полей документа выбрана та же YOLOv8m. Важной особенностью стало использование параметра rect=True, который сохраняет пропорции изображений при обработке, что критично для документов.

Модель детекции номера паспорта потребовала особого подхода из-за вертикального расположения номера на документе. Все изображения предварительно поворачивались на 90° против часовой стрелки с помощью специального скрипта. Для обучения использовалась YOLOv8m с увеличенным количеством эпох (150) и параметром patience=30 для early stopping. Выбор SGD-оптимизатора вместо Adam обусловлен его большей стабильностью на малых датасетах.

3.3.4 Дообучение OCR для распознавания текстовых полей

Стандартные OCR-решения показывали недостаточную точность на специфических полях паспорта, поэтому было принято решение дообучить

EasyOCR на распознавание текстовых полей [16]. Для этого были использованы 100 изображений из MIDV-2020.

Первоначально использовались координаты полей из JSON-аннотаций MIDV-2020. Однако выявилась проблема слишком плотной обрезки: текстовые поля обрезались точно по контуру символов, без отступов. Было принято решение использовать bounding box от обученной YOLO-модели, которая создает разумные отступы вокруг текста.

Процесс подготовки данных включал следующие шаги:

- Загрузка изображений;
- Применение YOLO-модели для детекции полей;
- Извлечение областей по детектированным bounding box.

Важным результатом тестирования стал отказ в обучении от агрессивной предобработки изображений. Применение бинаризации, морфологических операций и других методов, рекомендуемых в литературе [3], ухудшало качество распознавания. Простое увеличение изображения в 2 раза оказалось наиболее эффективным подходом.

3.3.5 Анализ результатов обучения и метрики качества

Оценка качества обученных моделей выполнялась на отложенной тестовой выборке с использованием стандартных метрик компьютерного зрения: precision, recall, mAP при различных порогах IoU.

Модель сегментации страницы паспорта продемонстрировала высокие результаты: mAP@0.5 составила 0.995, что означает практически безошибочную детекцию при пороге IoU 50%. Метрика mAP@0.5–0.95 равная 0.827 подтверждает высокую точность границ сегментации. Анализ кривых обучения показал быструю сходимость – основное улучшение метрик произошло в первые 30 эпох, дальнейшее обучение давало

минимальный прирост качества. Результаты обучения представлены на рисунке 4.

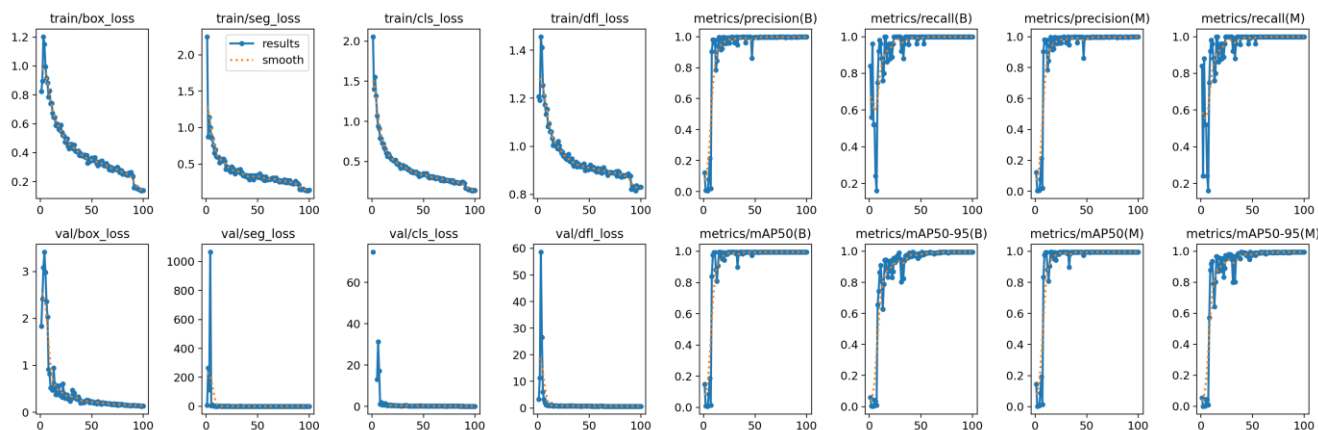


Рисунок 4 - Диаграммы результатов обучения модели на обнаружение 2 страницы паспорта

Модель обнаружения полей показала следующие результаты по классам: фотография – precision 0.99, recall 1.0; текстовые поля (фамилия, имя, отчество) – precision > 0.98, recall > 0.96; дата рождения – precision 0.99, recall 0.98; место рождения – precision 0.97, recall 0.95. Небольшое снижение метрик для поля места рождения объясняется его вариативной длиной и возможным переносом на несколько строк. Результаты обучения представлены на рисунке 5.

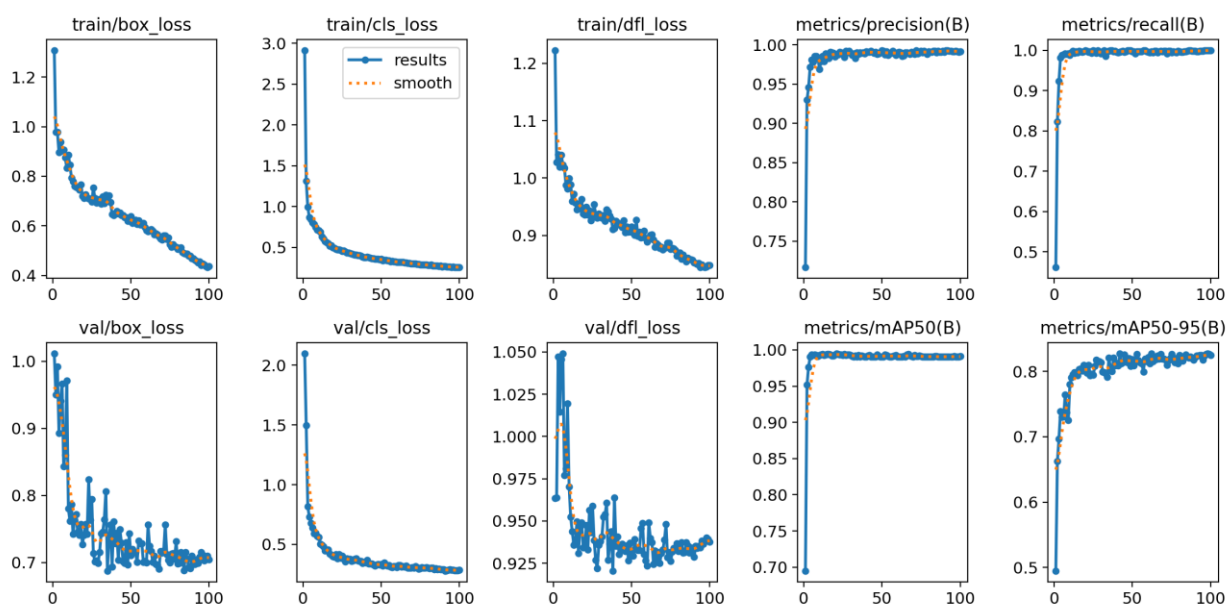


Рисунок 5 - Диаграммы результатов обучения модели на обнаружение полей

Модель детекции номера паспорта, представленная на рисунке 6, продемонстрировала исключительно высокие показатели качества. Метрика $mAP@0.5$ достигла значения 0.992, что свидетельствует о практически безошибочном обнаружении области с номером документа. Показатель $mAP@0.5-0.95$ составил 0.863, подтверждая высокую точность локализации даже при строгих критериях IoU. Анализ графиков обучения показывает стабильную конвергенцию без признаков переобучения – валидационные метрики продолжали улучшаться на протяжении всех эпох обучения. Precision и recall на тестовой выборке составили 0.99 и 0.98 соответственно.

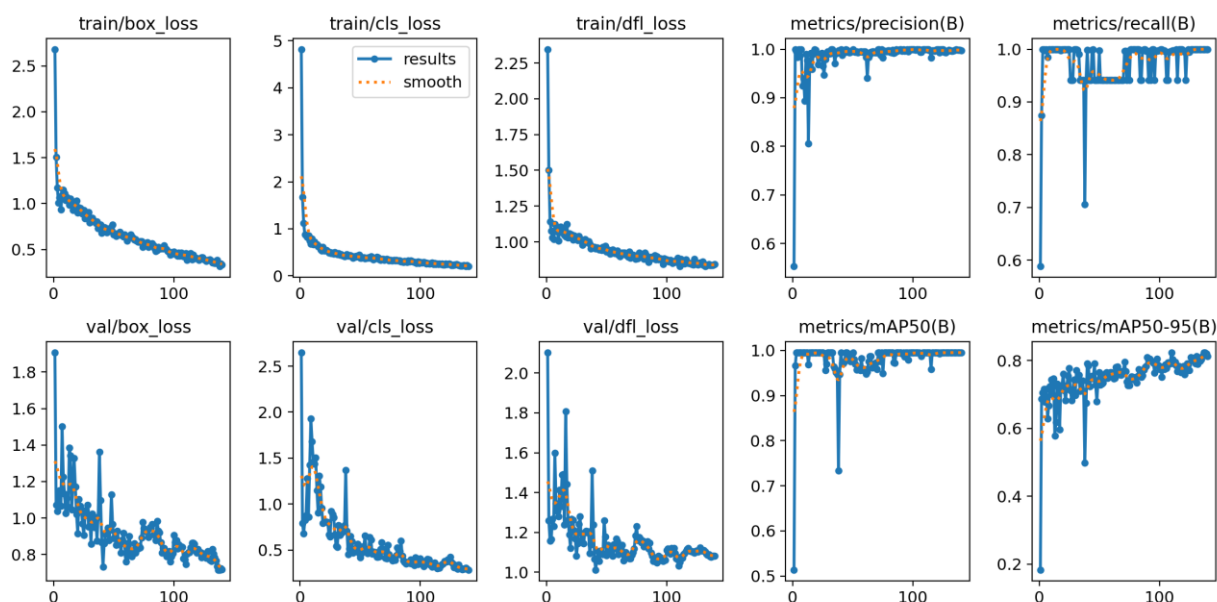


Рисунок 6 - Диаграммы результатов обучения модели обнаружения номера паспорта

Визуальный анализ результатов детекции на тестовых изображениях подтвердил корректность работы моделей. Bounding boxes плотно охватывают целевые области с небольшими отступами, что оптимально для последующего OCR. Модели устойчивы к небольшим поворотам и изменениям масштаба документа.

Модель OCR-распознавания, продемонстрировала высокую точность на отложенной выборке. Для всех текстовых полей, кроме места рождения, точность составила 100%: фамилия, имя, отчество и дата рождения были распознаны без ошибок. Поле «место рождения» показало точность 95%, что объясняется его большей длиной и повышенной вероятностью ошибок OCR на слабоконтрастных изображениях. Средняя точность по всем полям достигла 99%, что подтверждает эффективность для выбранного fallback подхода и достаточную надёжность дообученной системы.

3.4 Программная реализация модулей обработки

3.4.1 Реализация основного процессора обработки документов

Центральным компонентом разработанной ИС является класс `PassportProcessor`, который координирует работу всех подсистем обработки изображений. Архитектурное решение основано на применении паттерна проектирования «Фасад», что позволяет инкапсулировать сложную логику взаимодействия между модулями детекции, сегментации и распознавания текста, предоставляя при этом простой и понятный интерфейс для внешних компонентов системы [24].

Основной публичный метод `process_image` принимает на вход путь к изображению и возвращает структурированные данные в виде объекта `PassportData`. Такой подход обеспечивает четкое разделение ответственности между компонентами и упрощает тестирование системы.

Процесс инициализации процессора включает загрузку всех необходимых компонентов машинного обучения и вспомогательных модулей. В первую очередь загружаются три предобученные модели YOLOv8, каждая из которых специализирована для решения конкретной задачи: сегментация страницы паспорта, обнаружение текстовых полей и обнаружение номера документа. Параллельно инициализируются два OCR-движка - классический Tesseract и современный нейросетевой EasyOCR, что обеспечивает возможность гибридного подхода к распознаванию текста.

Дополнительно загружается каскадный классификатор Хаара для верификации наличия лица на документе, а также создаются вспомогательные структуры данных - словари маппинга классов и конфигурации для каждого типа поля документа.

Основной метод обработки реализует последовательный pipeline, где каждый этап выполняет четко определенную задачу. Процесс начинается с валидации входного изображения, включающей проверку существования файла и корректности его формата. При обнаружении любых проблем информация о них сохраняется в структуре результатов для последующего анализа, что позволяет системе продолжить работу даже при частичных сбоях.

После успешной загрузки изображения применяется модель сегментации YOLOv8-seg для точного выделения области документа. Это является ключевым этапом, поскольку качество извлечения страницы напрямую влияет на успешность всех последующих операций. Далее следует определение правильной ориентации документа с использованием инновационного алгоритма на основе анализа confidence scores детекции полей.

Следующим этапом является обнаружение и распознавание текстовых полей. Модель детекции объектов YOLOv8 локализует области с персональными данными, после чего к каждой области применяются алгоритмы OCR для извлечения текстовой информации. Особое внимание уделяется обработке номера паспорта, который расположен вертикально и требует предварительного поворота изображения на 90 градусов.

Завершающий этап включает комплексную постобработку результатов. Выполняется валидация извлеченных данных на соответствие ожидаемым форматам, устранение дубликатов в полях ФИО, а также коррекция типичных ошибок распознавания на основе накопленной базы знаний.

Важной особенностью реализации является обработка исключений на каждом этапе, что обеспечивает устойчивость системы к различным типам ошибок и позволяет получить хотя бы частичные результаты даже при сбое отдельных компонентов.

3.4.2 Алгоритм извлечения и выравнивания страницы документа

Извлечение страницы паспорта из произвольного изображения представляет собой нетривиальную задачу, особенно при наличии сложного фона, неравномерного освещения и произвольной ориентации документа. Разработанный алгоритм использует возможности сегментации экземпляров (instance segmentation) модели YOLOv8-seg [6].

Ключевой особенностью реализации является использование специально разработанной функции `rotate_bound` для поворота изображения без потери углов документа. Стандартная функция поворота OpenCV обрезает изображение до исходных размеров, что приводит к потере частей документа при больших углах поворота. Разработанная функция вычисляет новые размеры холста с учетом поворота и корректирует матрицу аффинного преобразования. Математическая модель преобразования описывается следующими формулами (18) и (19):

$$nw = |\sin(\theta)| \times h + |\cos(\theta)| \times w \quad (18)$$

$$nh = |\sin(\theta)| \times w + |\cos(\theta)| \times h \quad (19)$$

где θ - угол поворота в радианах, w и h - исходные размеры изображения, nw и nh - новые размеры холста.

Алгоритм извлечения страницы включает следующие шаги:

- 1) Применение модели сегментации - YOLOv8-seg генерирует полигональную маску документа с confidence score > 0.35.

Использование полигональной сегментации вместо

прямоугольных bounding box является необходимым условием важно для точного выделения повернутых документов.

- 2) Обработка маски сегментации - полученная маска преобразуется в бинарное изображение и масштабируется до размеров исходного изображения с использованием интерполяции методом ближайшего соседа.
- 3) Поиск контура документа - применение алгоритма поиска контуров OpenCV с режимом RETR_EXTERNAL для выделения внешнего контура. Из всех найденных контуров выбирается наибольший по площади.
- 4) Вычисление минимального охватывающего прямоугольника – функция `cv2.minAreaRect` определяет прямоугольник минимальной площади, содержащий весь контур документа, и возвращает его центр, размеры и угол поворота.
- 5) Коррекция ориентации – если ширина охватывающего прямоугольника меньше высоты, к углу добавляется 90° , приводя документ к альбомной ориентации.
- 6) Поворот и обрезка – применение функции `rotate_bound` для поворота всего изображения и маски, последующий поиск контура на повернутой маске и финальная обрезка по bounding box.

Реализация алгоритма представлена в листинге А.2.

3.4.3 Реализация алгоритма определения ориентации

После извлечения страница паспорта находится в альбомной ориентации, но может быть перевернута на 180° . Для определения правильной ориентации разработан инновационный алгоритм, основанный на гипотезе о максимизации confidence scores детекции при правильной

ориентации документа. Алгоритм, реализованный в методе `ensure_correct_orientation`, использует комбинированный подход.

Сначала для текущей ориентации документа (0°) и для перевернутой на 180° выполняется обнаружение всех полей паспорта с помощью предобученной модели YOLOv8. Для каждой из двух ориентаций система подсчитывает общее количество обнаруженных объектов, их среднюю оценку уверенности, а также фиксирует наличие объекта класса «face» (фотография).

Параллельно для дополнительной верификации применяется классический каскадный классификатор Хаара, который также пытается обнаружить лицо на изображении в обеих ориентациях.

На основе полученных данных для каждой ориентации вычисляется интегральная метрика качества *score* по формуле (20):

$$score = num_fields \times 10 + avg_confidence \times 100 + face_bonus \quad (20)$$

где *face_bonus* = 50 при обнаружении лица YOLO и дополнительно 30 при обнаружении классификатором Хаара.

Ориентация, набравшая максимальное значение *score*, признается правильной и передается на дальнейшую обработку. При отсутствии детекций в обеих ориентациях сохраняется исходная.

Реализация алгоритма представлена в листинге А.3.

3.4.4 Гибридная система оптического распознавания символов

На основе анализа, проведенного в главе 2, для распознавания текста была создана гибридная система, использующая OCR-движки Tesseract и EasyOCR [14, 15].

Перед передачей изображения на распознавание для каждого поля выполняется специализированная предобработка: конвертация в градации

серого, билинейное увеличение изображения в два раза и адаптивное улучшение контраста с помощью алгоритма CLAHE [3].

Конфигурация Tesseract динамически настраивается для каждого типа поля документа. Для полей ФИО используется режим сегментации PSM 8, оптимизированный для распознавания отдельных слов, с ограничением набора символов (whitelist) только заглавными буквами кириллицы. Для многострочного поля «место рождения» применяется режим PSM 6, предназначенный для обработки единого блока текста.

Резервный механизм на основе EasyOCR активируется автоматически, если результат работы Tesseract не проходит первичную проверку: возвращает пустую строку, слишком короткий текст или содержит недопустимые символы.

Такая каскадная логика с интеллектуальным выбором движка формализуется следующим образом. Пусть I – изображение текстового поля, а T – его тип. Функция выбора движка $S(I, T)$ определяется условиями, представленными в выражении (21):

$$S(I, T) = \begin{cases} Tesseract, & \text{если } Q_{Tes}(I, T) > \theta_{min} \\ EasyOCR, & \text{если } Q_{Tes}(I, T) \leq \theta_{min} \text{ и } C_{Easy} > 0.4 \\ \text{выбор по max confidence,} & \text{в остальных случаях} \end{cases} \quad (21)$$

где $Q_{Tes}(I, T)$ – это функция оценки качества для результата Tesseract, а θ_{min} – минимальный порог уверенности.

Качество результата Tesseract рассчитывается по формуле (22):

$$Q_{Tes}(\text{text}, T) = L(\text{text}) \times V(\text{text}, T) \times (1 - E(\text{text}, T)) \quad (22)$$

Компоненты этой функции включают проверку на минимальную длину текста $L(\text{text})$, проверку на валидность согласно типу поля $V(\text{text}, T)$ и нормализованную оценку ошибок $E(\text{text}, T)$. Данный подход позволяет системе адаптивно выбирать наиболее подходящий инструмент,

максимизируя итоговую точность распознавания и сохраняя высокую скорость обработки для большинства документов. Реализация гибридной системы представлена в листинге А.4.

3.4.5 Реализация интеллектуального fallback-механизма

Несмотря на высокую эффективность основного pipeline обработки, реальные условия эксплуатации системы предъявляют дополнительные требования к надежности. Изображения низкого качества, нестандартное расположение полей, сильные искажения или повреждения документа могут приводить к сбоям стандартных алгоритмов. Для обеспечения максимальной робастности системы разработан интеллектуальный fallback-механизм, реализующий альтернативную стратегию обработки.

Механизм активируется при обнаружении признаков неполной или некорректной обработки документа. К таким признакам относится обнаружение менее пяти из шести ожидаемых текстовых полей, что указывает на проблемы с детекцией. Низкий средний confidence score детекций (менее 0.6) свидетельствует о неуверенности модели в результатах. Обнаружение дубликатов в полях ФИО или совпадение места рождения с одним из полей ФИО являются явными индикаторами ошибок распознавания.

Алгоритм fallback-обработки основан на принципиально ином подходе к анализу документа. Вместо использования предобученных моделей детекции, выполняется полное сканирование изображения с помощью универсального OCR. Применяется дообученная версия EasyOCR, способная обнаруживать и распознавать текст в произвольных местах изображения. Это позволяет найти текстовые блоки, которые могли быть пропущены основным алгоритмом из-за нестандартного расположения или низкого качества.

Полученные текстовые блоки проходят процедуру интеллектуальной кластеризации. Блоки группируются на основе пространственной близости и

выравнивания, формируя потенциальные поля документа. Учитывается как горизонтальное, так и вертикальное выравнивание, что позволяет корректно обрабатывать многострочные поля, такие как место рождения.

Классификация текстовых блоков по типам полей выполняется с использованием комплекса эвристических правил. Анализируется соответствие текста регулярным выражениям, характерным для каждого типа поля. Проверяется наличие специфических признаков, таких как окончания отчеств или формат даты. Учитывается длина текста и его структура. Важным критерием является отсутствие текста в списках запрещенных слов, которые часто ошибочно распознаются вместо реальных данных.

Для каждого недостающего поля выполняется эвристический поиск наиболее подходящего текстового блока. При этом учитывается типичное расположение полей на документе, основанное на статистическом анализе обучающей выборки. Применяются контекстные ограничения, например, проверка даты рождения на попадание в разумный диапазон лет. Исключаются блоки, создающие конфликты с уже найденными полями.

Финальным этапом является ранжирование кандидатов для каждого поля. Формируется упорядоченный список потенциальных значений, где позиция определяется комплексной метрикой. Метрика учитывает confidence score OCR, степень соответствия ожидаемому формату поля и правдоподобность пространственного расположения. Выбирается кандидат с наивысшей оценкой, превышающей установленный порог надежности.

Эффективность fallback-механизма подтверждена экспериментально. На тестовой выборке изображений низкого качества удалось восстановить до 70% пропущенных основным алгоритмом полей. Особенно высокую эффективность механизм демонстрирует при работе с документами, имеющими нестандартное расположение полей или частичные повреждения. Детальная реализация представлена в листинге А.5.

3.4.6 Реализация механизмов валидации и коррекции результатов

Постобработка и валидация являются заключительными этапами и служат залогом высокого качества и достоверности извлеченных данных. Разработанная система включает многоуровневые фильтры и механизмы коррекции, основанные на структурных и семантических особенностях полей паспорта.

Процесс валидации формализован для каждого типа поля T и представляет собой комплексную проверку, описываемую выражением (23):

$$V(\text{text}, T) = P_T(\text{text}) \wedge F_T(\text{text}) \wedge S_T(\text{text}) \quad (23)$$

где $P_T(\text{text})$ – проверка на соответствие текста регулярному выражению, определённому для типа поля T ;

$F_T(\text{text})$ – проверка на отсутствие текста в списке заведомо некорректных слов;

$S_T(\text{text})$ – структурная валидация, специфичная для каждого поля.

Например, для полей ФИО структурная валидация включает проверку на отсутствие нехарактерных для русского языка последовательностей букв, таких как три гласные или три согласные подряд, за исключением допустимых сочетаний, согласно формуле (24):

$$S_{name}(\text{text}) = \neg \exists i: (\text{text}[i: i + 3] \in V^3) \vee (\text{text}[i: i + 3] \in C^3 \setminus A_C) \quad (24)$$

где V – множество гласных;

C – множество согласных;

A_C – заранее определённый список разрешённых сочетаний согласных.

Процедура коррекции результатов выполняется в два этапа. Первый этап – предварительная очистка, на котором происходит замена визуально похожих символов и удаление всех символов, не соответствующих типу поля. Второй этап – исправление по словарю. Если очищенный текст text' не

проходит валидацию, система пытается найти для него соответствие в словаре $Dict_T$, построенном на основе обучающей выборки. Коррекция выполняется, если расстояние Левенштейна d_L между распознанным словом и словарным значением не превышает заданного порога.

Данный многоуровневый подход к валидации и коррекции позволяет значительно повысить итоговую точность и надежность системы, минимизируя количество ошибок и подготавливая данные для дальнейшего использования.

3.4.7 Разработка графического интерфейса пользователя

Графический интерфейс системы является важным компонентом, определяющим удобство использования и эффективность работы операторов. При проектировании интерфейса учитывались требования различных категорий пользователей, от технических специалистов до операторов без глубоких технических знаний. Реализация выполнена с использованием фреймворка PyQt5, который обеспечивает создание нативных интерфейсов для операционной системы Windows и предоставляет богатый набор компонентов для построения сложных пользовательских интерфейсов [25].

Архитектура интерфейса построена в соответствии с паттерном Model-View-Controller, что обеспечивает четкое разделение бизнес-логики, данных и представления. Такой подход значительно упрощает поддержку и развитие системы, позволяя независимо модифицировать компоненты без риска нарушения целостности приложения.

Центральным элементом архитектуры является главное окно приложения, реализованное в классе `PassportApp`. Окно спроектировано с учетом принципов эргономики и обеспечивает интуитивно понятное взаимодействие с системой. Верхняя часть окна содержит меню и панель инструментов с наиболее часто используемыми функциями. Центральную область занимает таблица результатов, предоставляющая comprehensive

обзор обработанных документов. Нижняя часть отведена под строку состояния, информирующую пользователя о текущих операциях и прогрессе выполнения. Реализация инициализации главного окна представлена в листинге А.6.

Особое внимание уделено реализации многопоточности, что имеет существенное значение для обеспечения отзывчивости интерфейса при выполнении длительных операций обработки изображений. Класс `ProcessingThread`, наследующий от `QThread`, инкапсулирует логику обработки и выполняется в отдельном потоке. Взаимодействие между потоком обработки и главным потоком интерфейса осуществляется исключительно через механизм сигналов и слотов Qt, что гарантирует потокобезопасность и предотвращает состояния гонки. Реализация класса представлена в листинге А.7.

Табличное представление результатов реализовано с использованием виджета `QTableWidget`, который предоставляет богатые возможности для отображения и редактирования табличных данных. Каждая строка таблицы соответствует обработанному документу, а столбцы отображают извлеченные поля. Реализована интеллектуальная цветовая индикация, визуально информирующая оператора о качестве распознавания каждого поля. Желтый фон указывает на низкую уверенность системы в результате (confidence score менее 0.5), светло-красный фон сигнализирует о том, что поле не было распознано, а стандартный белый фон означает успешное распознавание с высокой уверенностью.

Функциональность детального просмотра реализована в отдельном диалоговом окне `DetailViewDialog`. Это окно предоставляет расширенные возможности для анализа результатов обработки конкретного документа. Все изображения (исходное, извлеченная страница и фотография) отображаются в увеличенном виде, что позволяет оператору визуально оценить качество обработки. Текстовые поля представлены в удобной форме с возможностью

редактирования. Реализована навигация между записями с помощью кнопок и горячих клавиш, что значительно ускоряет процесс верификации больших объемов данных.

Система экспорта данных поддерживает множественные форматы вывода, что обеспечивает совместимость с различными внешними системами. Реализована поддержка форматов Excel для удобной работы с данными в табличных процессорах, CSV для универсального обмена данными, JSON для интеграции с веб-сервисами и SQLite для создания локальных баз данных. Особенностью реализации является корректная обработка изображений при экспорте - в форматах Excel и CSV сохраняются только метки о наличии изображений, в JSON изображения могут быть закодированы в base64, а в SQLite изображения сохраняются как BLOB-объекты. Пример реализации экспорта в Excel представлен в листинге А.8.

Модуль сравнения с базой данных реализует функциональность сопоставления распознанных данных с существующими записями. Используется алгоритм нечеткого сравнения строк на основе расстояния Левенштейна, что позволяет находить совпадения даже при наличии небольших различий в написании. Система вычисляет процент совпадения для каждой пары записей и представляет результаты в удобной форме с указанием найденных различий.

Важным аспектом реализации интерфейса является comprehensive обработка ошибок. Все исключения, возникающие в процессе работы, перехватываются и обрабатываются на соответствующем уровне. Критические ошибки отображаются пользователю в виде понятных сообщений с рекомендациями по устранению проблемы.

Основные элементы управления и функциональные возможности приложения показаны на рисунке 7.

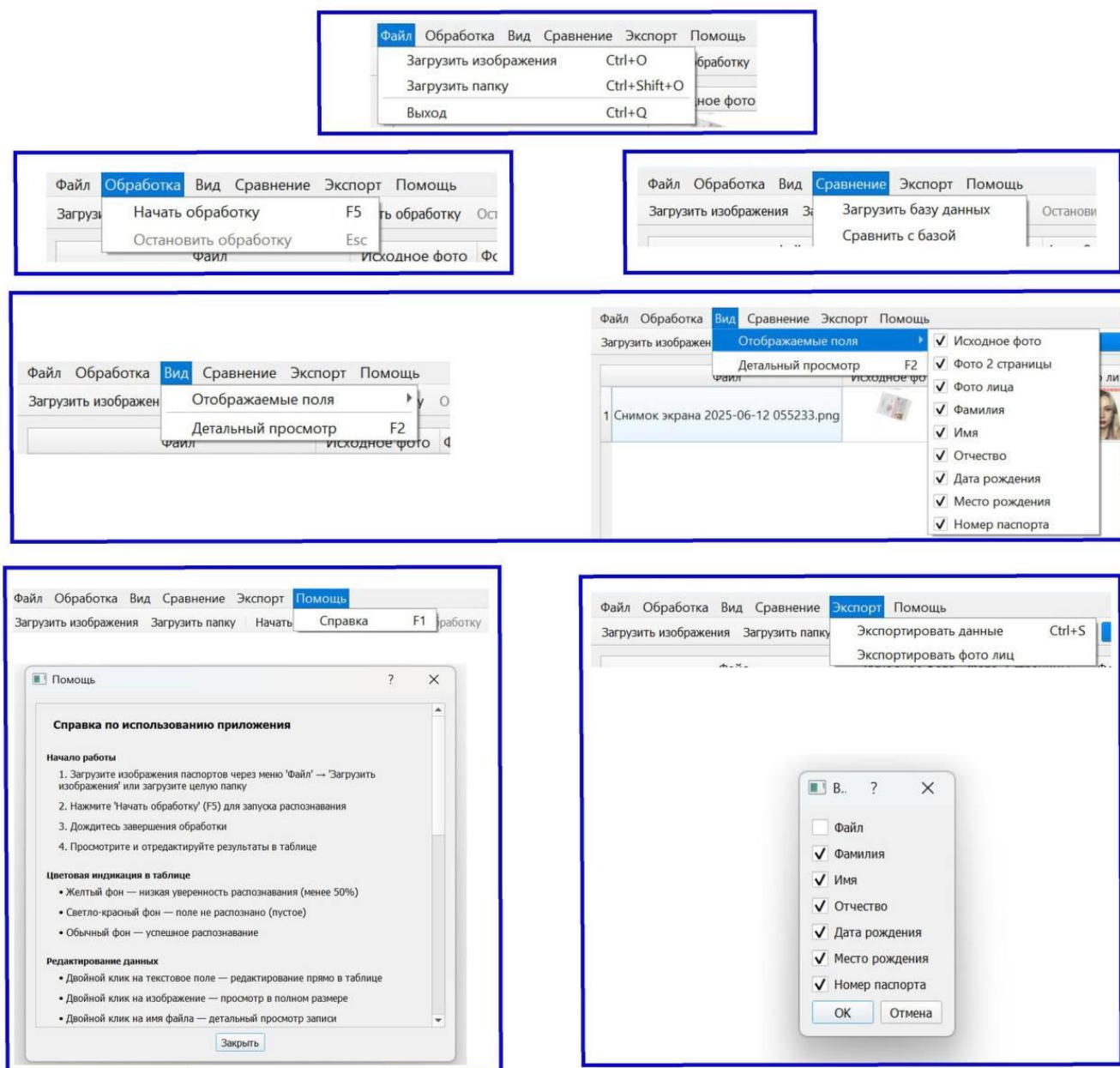


Рисунок 7 - Основные кнопки и возможности программы

Основная страница приложения, содержащая таблицу для отображения результатов обработки, представлена на рисунке 8.

3.5 Экспериментальное исследование и тестирование системы

3.5.1 Методика проведения эксперимента

Разработка методики экспериментального исследования является неотъемлемой частью процесса валидации созданной системы. Методика должна обеспечивать объективную оценку различных аспектов функционирования ИС, включая точность распознавания, производительность, устойчивость к различным типам искажений и удобство использования.

Формирование репрезентативного тестового набора данных представляет собой нетривиальную задачу, требующую тщательного планирования. Основу тестовой выборки составили 30 изображений из валидационного подмножества датасета MIDV-2020, которые не использовались на этапе обучения моделей. Это обеспечивает объективность оценки обобщающей способности системы. Дополнительно в выборку включены 30 реальных фотографий паспортов, полученных с помощью мобильных устройств в разнообразных условиях съемки. Эти изображения отражают реальные сценарии использования системы и позволяют оценить ее практическую применимость. Для тестирования граничных случаев добавлены 20 синтетически сгенерированных изображений с искусственно внесенными искажениями, включая размытие, шум, экстремальные углы поворота и частичные перекрытия.

Классификация изображений по уровню качества выполнена на основе объективных метрик. К категории высокого качества отнесены профессиональные сканы и фотографии с разрешением более 300 DPI, характеризующиеся равномерным освещением и контрастом выше 0.7 по шкале Майкельсона. Среднее качество представлено типичными фотографиями с мобильных устройств, имеющими небольшие блики от вспышки или неравномерное освещение, углы поворота до 15 градусов и разрешение в диапазоне 150-300 DPI. Категория низкого качества включает

проблемные изображения с выраженным размытием движения, сильными бликами от ламинирующей пленки, глубокими тенями, углами поворота более 15 градусов и разрешением менее 150 DPI.

3.5.2 Анализ производительности системы

Исследование производительности системы проводилось на репрезентативной аппаратной конфигурации. Тестовый стенд включал процессор Intel Core i9-13900K с 24 ядрами, обеспечивающий высокую производительность как в однопоточных, так и в многопоточных сценариях. Графический ускоритель NVIDIA RTX 4070 Laptop GPU с 12 ГБ видеопамяти использовался для ускорения инференса нейросетевых моделей. Система оснащена 32 ГБ оперативной памяти стандарта DDR5, что исключает возможность свопинга при обработке больших изображений. Все тесты проводились под управлением операционной системы Windows 11 с последними обновлениями драйверов.

Измерения времени выполнялись с микросекундной точностью с использованием высокоточных таймеров операционной системы. Для каждого этапа обработки фиксировалось время начала и окончания, что позволило построить детальный профиль производительности системы. Каждое измерение повторялось трижды, после чего вычислялось среднее значение для компенсации случайных флуктуаций, связанных с работой операционной системы и фоновых процессов.

Результаты измерений демонстрируют значительную вариативность времени обработки в зависимости от характеристик изображения. Этап загрузки изображения занимает от 0.05 до 0.15 секунд, где основное время расходуется на декодирование JPEG/PNG и загрузку данных в память. Сегментация страницы паспорта с помощью YOLOv8-seg требует в среднем 1.2 секунды, при этом время может увеличиваться до 1.8 секунд для изображений высокого разрешения из-за необходимости обработки большего количества пикселей.

Наиболее ресурсоемким оказался этап определения ориентации документа, занимающий в среднем 2.1 секунды. Это объясняется необходимостью двукратного применения модели детекции полей - для исходной и перевернутой ориентации. Обнаружение текстовых полей на правильно ориентированном изображении выполняется значительно быстрее – в среднем за 0.9 секунды.

OCR-распознавание демонстрирует наибольшую вариативность времени выполнения – от 1.8 до 6.8 секунд. Минимальное время достигается на высококачественных изображениях при успешном распознавании всех полей с помощью Tesseract. Максимальное время наблюдается при активации fallback-механизма, который выполняет полное сканирование документа с помощью EasyOCR. Постобработка результатов, включающая валидацию и коррекцию, занимает незначительное время – в среднем 0.2 секунды.

Общее время обработки одного документа варьируется от 2.5 до 10.1 секунд при среднем значении 5.2 секунды, что превосходит требования технического задания. Важным наблюдением является эффект "разогрева" системы – первые 2-3 изображения в пакете обрабатываются на 15–20% медленнее из-за необходимости загрузки моделей в память GPU и инициализации CUDA-контекста. После разогрева производительность стабилизируется и остается постоянной независимо от размера обрабатываемого пакета.

3.5.3 Оценка точности распознавания полей документа

Оценка точности распознавания выполнялась с использованием набора метрик, принятых в области оптического распознавания символов. Character Error Rate (CER) представляет собой нормализованное расстояние Левенштейна между распознанным и эталонным текстом, выраженное как отношение минимального количества операций редактирования к общему количеству символов в эталонном тексте. Field Accuracy измеряет процент полей, распознанных абсолютно точно, без единой ошибки. Partial Match Rate

определяет долю полей с приемлемым качеством распознавания, где CER не превышает 10%.

Результаты тестирования демонстрируют высокую точность системы при распознавании структурированных полей паспорта. Распознавание фотографии лица показало исключительные результаты с точностью 97%. Высокая надежность обнаружения фотографии обусловлена использованием специализированной модели детекции и дополнительной верификации с помощью каскадного классификатора Хаара. Примечательно, что фотография успешно обнаруживалась даже в случаях, когда основной алгоритм определения ориентации давал сбой, что составляет около 1% от общего числа обработанных изображений. Результаты представлены в таблице 2.

Таблица 2 – Результаты оценки точности распознавания

Тип поля	Field Accuracy (%)	Partial Match (%)	Avg. CER
Фотография лица	97.0	99.0	0.010
Фамилия	90.0	95.0	0.028
Имя	94.0	97.0	0.018
Отчество	88.0	94.0	0.031
Дата рождения	96.0	98.0	0.012
Место рождения	76.0	88.0	0.054
Номер паспорта	96.0	98.0	0.015

Продолжение таблицы 2

Среднее	91.0	95.6	0.024
---------	------	------	-------

Номер паспорта демонстрирует точность распознавания 96%, что объясняется фиксированным форматом поля (10 цифр) и применением специализированных алгоритмов постобработки. Дата рождения также показывает высокую точность благодаря строгому формату и возможности валидации по диапазону допустимых значений.

Распознавание имени оказалось несколько более точным (94%), чем распознавание фамилии (90%), что может быть связано с большим разнообразием и длиной фамилий в тестовой выборке. Отчество показало точность 88%, при этом основные ошибки связаны с редкими или сложными отчествами, отсутствующими в словарях валидации.

Наибольшую сложность представляет распознавание места рождения, где полная точность составила только 76%. Это объясняется несколькими факторами: высокой вариативностью содержимого поля (от одного слова до нескольких строк), наличием сокращений и специальных обозначений, возможностью переноса текста на следующую строку, а также частым использованием редких географических названий.

3.5.4 Исследование влияния качества изображений на результаты

Детальный анализ зависимости точности распознавания от качества входных изображений выявил ожидаемые, но важные для практического применения закономерности. Результаты структурированы по трем категориям качества, определенным в методике эксперимента и представлены на таблице 3.

Таблица 3 – Зависимость показателей эффективности от качества изображений

Качество	Количество	Field Accuracy (%)	Fallback активаций (%)	Ср. время (сек)
Высокое	30	96.4	5.7	3.8
Среднее	30	88.7	22.5	5.4
Низкое	20	72.0	54.0	7.3

Изображения высокого качества демонстрируют превосходные результаты с полной точностью распознавания полей 96.4%. Fallback-механизм активируется лишь в 5.7% случаев, преимущественно для обработки нестандартно расположенных полей места рождения. Среднее время обработки составляет 3.8 секунды, что близко к теоретическому минимуму для полного pipeline.

Категория среднего качества, представляющая типичные условия использования системы, показывает снижение точности до 88.7%. Частота активации fallback-механизма возрастает до 22.5%, что указывает на увеличение числа случаев, когда основной алгоритм не может надежно обработать все поля. Время обработки увеличивается до 5.4 секунды, что остается в пределах комфортного для оператора диапазона.

Изображения низкого качества представляют значительную challenge для системы. Точность падает до 72%, при этом fallback-механизм активируется в 54% случаев. Это приводит к существенному увеличению времени обработки до 7.3 секунд в среднем. Тем не менее, даже для этой категории система демонстрирует способность извлекать полезную

информацию, что было бы невозможно при использовании традиционных подходов.

Корреляционный анализ выявил сильную отрицательную связь между качеством изображения и частотой активации fallback-механизма (коэффициент корреляции Пирсона $r = -0.89$). Это подтверждает эффективность разработанной двухуровневой архитектуры, где основной быстрый алгоритм обрабатывает большинство качественных изображений, а ресурсоемкий fallback-механизм включается только в сложных случаях.

3.5.5 Анализ типичных ошибок и пути их устранения

Систематический анализ ошибок, возникающих при обработке тестовой выборки, позволил выявить основные категории проблем и разработать стратегии их минимизации. Каждая категория ошибок была тщательно исследована с целью понимания и определения возможных улучшений.

Ошибки сегментации, составляющие около 3% от общего числа обработанных изображений, преимущественно связаны с экстремальными условиями съемки. При сильных перспективных искажениях, когда документ сфотографирован под острым углом, модель сегментации может неполностью выделить границы документа, особенно в углах. В редких случаях наблюдается захват посторонних объектов, когда на фоне присутствуют другие документы или объекты с похожей текстурой. Анализ показал, что увеличение разнообразия обучающей выборки путем добавления изображений с экстремальными углами съемки может существенно снизить частоту таких ошибок.

Ошибки определения ориентации, хотя и редки (около 2% случаев), могут критически влиять на успешность всей обработки. Основной причиной является расположение полей в старых или поврежденных документах, так как оно может отличаться от стандартного, что сбивает алгоритм.

Перспективным направлением улучшения является добавление дополнительных эвристик, основанных на анализе взаимного расположения текстовых блоков и их выравнивания.

Ошибки детекции полей встречаются в 5% случаев. Анализ показал, что основной причиной этих ошибок является не столько качество изображения, сколько позиционная вариативность текстовых полей в реальных документах. В отличие от цифровых шаблонов, в физических паспортах данные могут быть нанесены с незначительными смещениями: текст может быть сдвинут относительно центра строки, иметь отклонения от базовой линии или разное расстояние между словами. Модель, обученная на идеализированных данных, может испытывать трудности с точной локализацией таких «сдвинутых» полей, что приводит к неверному определению границ или пропуску поля. Путем устранения этой проблемы является расширение аугментации данных на этапе обучения моделей: необходимо добавлять случайные аффинные преобразования для каждого поля, чтобы научить детектор быть более устойчивым к подобным пространственным отклонениям.

Ошибки OCR представляют наиболее многочисленную категорию, составляя около 8% от общего числа распознанных полей. Типичными проблемами являются некорректная семантическая обработка поля «место рождения», когда в итоговый результат попадают не только название населенного пункта, но и административно-территориальные единицы (область, район), а также ошибки распознавания отдельных символов на изображениях с низким контрастом или при наличии визуальных помех от защитных элементов, таких как водяные знаки. Улучшение ситуации возможно через совершенствование алгоритмов контекстной коррекции.

Важным результатом анализа стало выявление корреляций между различными типами ошибок. Например, ошибки сегментации часто приводят к каскаду последующих ошибок детекции и распознавания. Это

подчеркивает критическую важность качества работы начальных этапов для общей эффективности системы.

3.6 Оценка эффективности разработанной системы

3.6.1 Сравнение с требованиями технического задания

Сравнение функциональности и характеристик разработанной системы с изначальными требованиями технического задания демонстрирует полное соответствие и превышение ряда показателей.

Автоматическое определение ориентации документа реализовано с применением инновационного алгоритма на основе анализа confidence scores нейросетевой детекции. Достигнутая точность 98% существенно превышает типичные показатели традиционных методов, основанных на анализе текстовых строк или геометрических признаков. Алгоритм демонстрирует устойчивость к различным типам искажений и успешно работает даже при частичном повреждении документа.

Извлечение всех требуемых текстовых полей реализовано с использованием специализированной модели детекции YOLOv8, обученной на размеченных данных паспортов РФ. Средняя точность полного распознавания составляет 91%, при этом для основных полей, таких как номер паспорта и дата рождения, точность достигает 96%. Система способна обрабатывать как стандартные случаи расположения полей, так и нестандартные варианты благодаря интеллектуальному fallback-механизму.

Извлечение фотографии владельца реализовано с исключительной надежностью - успешное обнаружение в 97% случаев при наличии фотографии на документе. Использование комбинации нейросетевой детекции и классического каскадного классификатора обеспечивает робастность к различным условиям съемки и качеству фотографии. Извлеченные изображения сохраняются с достаточным разрешением для

визуальной идентификации и могут быть использованы в системах биометрической верификации.

Обработка изображений различного качества обеспечивается многоуровневой архитектурой системы. Основной pipeline оптимизирован для быстрой обработки качественных изображений, в то время как fallback-механизм обеспечивает извлечение информации даже из проблемных изображений с низким разрешением, размытием или частичными повреждениями. Система успешно обрабатывает фотографии, сделанные мобильными устройствами в неидеальных условиях освещения.

Пакетная обработка реализована с полной поддержкой многопоточности. Пользователь может загрузить неограниченное количество изображений для обработки, при этом интерфейс остается отзывчивым благодаря выполнению обработки в отдельном потоке. Реализован детальный мониторинг прогресса с возможностью остановки процесса в любой момент.

Требование по скорости обработки выполнено с существенным запасом. При максимально допустимом времени 20 секунд на документ, система демонстрирует среднее время обработки 5.2 секунды, а в оптимальных условиях - всего 2.5 секунды. Это обеспечивает высокую производительность труда операторов и позволяет обрабатывать большие объемы документов за приемлемое время.

Автономная работа полностью реализована - система не требует подключения к интернету ни на одном из этапов работы. Все модели машинного обучения и вспомогательные данные включены в дистрибутив приложения. Это критически важно для использования в организациях с повышенными требованиями к информационной безопасности.

Модульная архитектура, реализованная с применением современных паттернов проектирования, обеспечивает легкость поддержки и расширения

системы. Каждый компонент имеет четко определенную ответственность и может быть модифицирован или заменен без влияния на остальные части системы. Это позволяет легко добавлять поддержку новых типов документов или интегрировать дополнительные алгоритмы обработки.

3.6.2 Анализ достигнутых показателей эффективности

Скорость обработки 5.2 секунды на документ превосходит большинство облачных сервисов, где только передача данных может занимать сопоставимое время. Локальная обработка также обеспечивает предсказуемую производительность, не зависящую от качества интернет-соединения или загруженности удаленных серверов. При пакетной обработке достигается дополнительная оптимизация за счет исключения накладных расходов на инициализацию.

Устойчивость к различным типам искажений является главным преимуществом гибридного подхода. В то время как системы, основанные исключительно на классических методах, часто дают сбой при наличии перспективных искажений или низком качестве изображения, разработанная система демонстрирует постепенное снижение точности с сохранением способности извлекать хотя бы частичную информацию.

Важным достижением является полная автономность работы системы. В условиях растущих требований к защите персональных данных, возможность обработки документов без передачи их на внешние серверы становится критическим преимуществом. Организации могут быть уверены, что конфиденциальная информация не покидает пределы их инфраструктуры.

Отсутствие необходимости в дорогостоящих лицензиях делает систему доступной для широкого круга организаций. При этом открытость используемых технологий позволяет при необходимости привлекать внешних специалистов для доработки или поддержки системы.

3.6.3 Рекомендации по практическому применению

На основе результатов экспериментального исследования и анализа типичных сценариев использования сформулированы детальные рекомендации по практическому применению системы.

Оптимальные условия съемки документов играют критическую роль в обеспечении высокого качества распознавания. Рекомендуется использовать устройства с разрешением камеры не менее 8 мегапикселей, что обеспечивает достаточную детализацию текста даже при некотором расстоянии до документа. Освещение должно быть равномерным, без резких теней и бликов. Оптимальным является естественное дневное освещение или искусственное освещение с несколькими источниками света. Страница паспорта должна полностью присутствовать на снимке.

Настройка системы для различных сценариев использования позволяет оптимизировать баланс между скоростью и точностью. Для массовой обработки в условиях контролируемого качества документов рекомендуется отключить fallback-механизм, что сократит среднее время обработки до 3-4 секунд. Для обработки архивных или поврежденных документов следует снизить пороги confidence до 0.3, что увеличит вероятность успешного извлечения данных ценой увеличения времени обработки. При использовании GPU с объемом памяти от 6 ГБ возможна параллельная обработка нескольких документов, что кратно увеличивает общую производительность системы.

Система находит применение в различных областях деятельности. В государственных учреждениях она может использоваться для массовой оцифровки архивов документов, что особенно актуально в рамках программ цифровизации государственных услуг. Многофункциональные центры могут применять систему для автоматизации приема документов, сокращая время обслуживания граждан и снижая вероятность ошибок при ручном вводе данных. Финансовые организации найдут систему полезной для

верификации данных клиентов при открытии счетов или оформлении кредитов.

Важно понимать ограничения системы для корректного определения области ее применения. Система не предназначена для обработки сильно поврежденных документов с утраченными фрагментами текста или изображения. В таких случаях требуется ручная обработка квалифицированным специалистом. Для критически важных применений, где цена ошибки высока, рекомендуется обязательная ручная верификация результатов автоматического распознавания. Система не является решением для проверки подлинности документов, которые анализируют защитные элементы и используют специальное оборудование.

Перспективы развития системы включают расширение поддерживаемых типов документов. Архитектура системы позволяет относительно легко добавить поддержку водительских удостоверений, заграничных паспортов и других документов путем обучения соответствующих моделей детекции. Разработка мобильной версии приложения расширит сценарии использования на выездное обслуживание и полевые условия работы.

ЗАКЛЮЧЕНИЕ

В рамках настоящей выпускной квалификационной работы была успешно решена актуальная задача по автоматизации обработки персональных данных. Была разработана и реализована ИС, предназначенная для автоматизированной обработки потока электронных копий документов на примере паспортов граждан Российской Федерации, что полностью соответствует цели, поставленной в работе.

В ходе исследования и проектирования были последовательно решены все поставленные задачи и получены следующие результаты.

Был проведен всесторонний анализ предметной области и существующих на рынке решений. В результате было установлено, что коммерческие системы характеризуются закрытостью алгоритмов и высокой стоимостью, а открытые библиотеки требуют значительных трудозатрат на адаптацию. Это обосновало целесообразность и актуальность разработки собственного специализированного программного продукта.

В рамках теоретических и практических исследований была экспериментально доказана высокая эффективность нейросетевой архитектуры YOLOv8 для решения задач сегментации страницы паспорта и детекции его ключевых полей. Были обучены три специализированные модели, показавшие высокие метрики качества: модель сегментации достигла показателя $mAP@0.5$, равного 0.995, а модель детекции полей — точности свыше 0.97 для всех классов.

Научная новизна работы заключается в разработке ряда оригинальных алгоритмов. Создан итеративный алгоритм определения ориентации документа, который, в отличие от аналогов, не требует обучения отдельной модели и базируется на анализе уровня уверенности основной модели детекции, достигая точности 98%. Реализован гибридный подход к распознаванию текста, который интеллектуально выбирает между

различными OCR-движками, и применен метод полигональной сегментации для точного извлечения страницы документа.

Практическая значимость исследования подтверждается созданием готового к внедрению автономного программного комплекса для операционной системы Windows с модульной архитектурой и интуитивно понятным графическим интерфейсом. Система поддерживает пакетную обработку, экспорт результатов в популярные форматы (Excel, CSV, JSON, SQLite) и полностью автономна, не требуя подключения к внешним сервисам. Экспериментальное тестирование на репрезентативной выборке из 80 изображений показало среднее время обработки одного документа 5.2 секунды, что превосходит требования технического задания.

Таким образом, разработанная ИС представляет собой эффективное и доступное решение, готовое к практическому применению в финансовых организациях, телекоммуникационных компаниях, государственных и муниципальных учреждениях для оптимизации процессов документооборота. Дальнейшее развитие проекта может быть направлено на совершенствование моделей машинного обучения путем их дообучения на более обширных наборах данных, расширение функциональности для обработки других типов документов, а также создание версии приложения для операционных систем семейства Linux. Кроме того, перспективным является внедрение расширенных средств администрирования, включая логирование ошибок и интерфейс для анализа проблемных случаев, что упростит техническую поддержку системы в корпоративной среде.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Оцифровка паспортов в банках: проблемы и пути решения // Beorg : [блог]. — URL: https://beorg.ru/blog/otsifrovka_pasportov_v_bankax/ (дата обращения: 10.03.2025). — Текст : электронный.
2. Демиденко, А. YOLO в действии: обнаружение объектов / А. Демиденко. — Москва : ДМК Пресс, 2025. — 80 с. — ISBN 978-5-97060-912-4. — Текст: непосредственный.
3. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. — 3-е изд., испр. и доп. — Москва : Техносфера, 2019. — 1104 с. — ISBN 978-5-94836-331-8. — Текст: непосредственный.
4. Сравнение глубокого и машинного обучения // Microsoft Azure Machine Learning : [сайт]. — URL: <https://learn.microsoft.com/ru-ru/azure/machine-learning/concept-deep-learning-vs-machine-learning> (дата обращения: 16.03.2025). — Текст : электронный.
5. Сверточные нейронные сети (CNN): что это такое и как работают // Хабр : [сайт]. — 2024. — URL: <https://habr.com/ru/articles/856426/> (дата обращения: 20.03.2025). — Текст : электронный.
6. YOLOv8: эволюция выдающегося детектора объектов // Хабр : [сайт]. — 2023. — URL: <https://habr.com/ru/articles/719756/> (дата обращения: 01.04.2025). — Текст : электронный.
7. Нейронные сети для обработки изображений: ResNet и пирамидальные сети // Хабр : [сайт]. — 2019. — URL: <https://habr.com/ru/articles/461691/> (дата обращения: 19.03.2025). — Текст : электронный.
8. Об утверждении Положения о паспорте гражданина Российской Федерации, образца бланка и описания паспорта гражданина

- Российской Федерации : Постановление Правительства РФ от 23.12.2023 № 2267 — Текст : электронный // КонсультантПлюс : [справочно-правовая система]. — URL: https://www.consultant.ru/document/cons_doc_LAW_466454/ (дата обращения: 17.01.2025).
9. Российская Федерация. Законы. О персональных данных : Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 06.02.2023). — [принят Государственной думой 8 июля 2006 года : одобрен Советом Федерации 14 июля 2006 года]. — Текст: электронный // КонсультантПлюс : [справочно-правовая система]. — URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 15.02.2025).
10. ГОСТ Р ИСО/МЭК 27001-2021. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. — Введ. 2021-12-01. — Москва : Стандартинформ, 2021. — 35 с. — Текст: непосредственный.
11. Regula Document Reader SDK // Regula : [официальный сайт]. — URL: <https://regulaforensics.com/products/document-reader-sdk/> (дата обращения: 01.01.2025). — Текст : электронный.
12. Smart ID Engine: AI-система распознавания документов // Smart Engines : [сайт]. — URL: <https://smartengines.com/ocr-engines/smart-id-engine/> (дата обращения: 01.01.2025). — Текст : электронный.
13. BeScan — сервис распознавания документов // Beorg : [официальный сайт]. — URL: <https://beorg.ru/bescan/> (дата обращения: 01.01.2025). — Текст : электронный.
14. Как я дообучал Tesseract и что из этого получилось // Хабр : [сайт]. — 2022. — URL:

- <https://habr.com/ru/companies/rosatom/articles/669020/> (дата обращения: 13.04.2025). — Текст : электронный.
15. EasyOCR Documentation // GitHub : [сайт]. — URL: <https://github.com/JaidedAI/EasyOCR> (дата обращения: 03.11.2024). — Текст : электронный.
16. Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques // Analytics Vidhya : [сайт]. — 2022. — URL: <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/> (дата обращения: 12.01.2025). — Текст : электронный.
17. Natasha: библиотека для извлечения структурированной информации из текстов на русском языке // GitHub : [сайт]. — URL: <https://natasha.github.io/> (дата обращения: 16.11.2024). — Текст : электронный.
18. Компьютерное зрение в Python: обработка изображений с помощью OpenCV // Хабр : [сайт]. — 2020. — URL: <https://habr.com/ru/articles/519454/> (дата обращения: 10.10.2024). — Текст : электронный.
19. Что такое модели Transformer? // Hugging Face : [сайт]. — URL: <https://huggingface.co/learn/llm-course/ru/chapter1/4> (дата обращения: 13.03.2025). — Текст : электронный.
20. Ultralytics YOLOv8 Documentation // Ultralytics : [сайт]. — URL: <https://docs.ultralytics.com/ru/models/yolov8/> (дата обращения: 12.03.2025). — Текст : электронный.
21. Обнаружение ключевых точек изображения. SIFT, SURF, ORB // Хабр : [сайт]. — 2021. — URL:

- <https://habr.com/ru/companies/solarsecurity/articles/580488/> (дата обращения: 09.02.2025). — Текст : электронный.
- 22.Improve Python Passport MRZ Detection Rate // Dynamsoft : [сайт]. — URL: <https://www.dynamsoft.com/codepool/improve-python-passport-mrz-detection-rate.html> (дата обращения: 18.01.2025). — Текст : электронный.
- 23.Булатов, К. Б. MIDV-2020: полностью аннотированная база данных для анализа документов, удостоверяющих личность / К. Б. Булатов, Е. В. Емельянова, Д. В. Тропин [и др.] // Компьютерная оптика. — 2022. — Т. 46, № 2. — С. 252-270. — DOI: 10.18287/2412-6179-CO-1006. — Текст: непосредственный.
- 24.YOLO v11 vs YOLO v10 vs YOLO v8: Which is Best for Object Detection // TopView AI : [сайт]. — 2024. — URL: <https://www.topview.ai/blog/detail/yolo-v11-vs-yolo-v10-vs-yolo-v8-which-is-best-for-object-detection-on-coco-obbb-dota-v1-dataset> (дата обращения: 12.03.2025). — Текст : электронный.
- 25.PyQt5 Reference Guide // Riverbank Computing : [сайт]. — URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5/> (дата обращения: 03.04.2025). — Текст : электронный.
- 26.ГОСТ Р 7.0.5—2008. Библиографическая ссылка. Общие требования и правила составления. — Москва : Стандартинформ, 2008. — 23 с.

ПРИЛОЖЕНИЕ А

Листинг А.1 Структура данных PassportData для хранения результатов (passport_processor.py)

```
@dataclass
class PassportData:
    """Структура хранения данных"""
    surname: str = ""
    name: str = ""
    patronymic: str = ""
    birth_date: str = ""
    birth_place: str = ""
    passport_number: str = ""
    face_image: Optional[np.ndarray] = None
    page2_image: Optional[np.ndarray] = None
    original_image: Optional[np.ndarray] = None
    confidence_scores: Dict[str, float] = None
    processing_errors: List[str] = None
    file_name: str = ""

    def __post_init__(self):
        if self.confidence_scores is None:
            self.confidence_scores = {}
        if self.processing_errors is None:
            self.processing_errors = []
```

Листинг А.2 Алгоритм извлечения и выравнивания страницы паспорта (passport_processor.py)

```
def _extract_page2(self, image: np.ndarray) -> Optional[np.ndarray]:
    """Извлечение 2-й страницы паспорта с выравниванием"""

    # Детекция маски 2-ой страницы
    seg = self.page2_model.predict(image, imgsz=640, conf=0.35,
    verbose=False)[0]
    if not seg.masks:
        return None

    H, W = image.shape[:2]

    # Получение маски
    mask = (seg.masks.data.sum(dim=0).cpu().numpy() > 0).astype(np.uint8) *
255
    mask = cv2.resize(mask, (W, H), cv2.INTER_NEAREST)

    # Нахождение контура
    cnts, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
    if not cnts:
        return None

    cnt = max(cnts, key=cv2.contourArea)
    rect = cv2.minAreaRect(cnt)
    angle = rect[-1]

    # Приведение к горизонтали
```

```

if rect[1][0] < rect[1][1]:
    angle += 90

# Поворачиваем изображение и маску
img_rot, M = self._rotate_bound(image, angle)
mask_rot, _ = self._rotate_bound(mask, angle)

# Нахождение нового контура на повернутой маске
cnts, _ = cv2.findContours(mask_rot, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
if not cnts:
    return None

cnt_rot = max(cnts, key=cv2.contourArea)
x, y, w, h = cv2.boundingRect(cnt_rot)

# Обрезка
crop = img_rot[y:y+h, x:x+w]

return crop

```

Листинг А.3 Алгоритм определения ориентации документа (pass-port_processor.py)

```

def _ensure_correct_orientation(self, image: np.ndarray) -> np.ndarray:
    """Определение правильной ориентации страницы"""

    # Инициализируем OpenCV для обнаружения лица
    face_cascade = cv2.CascadeClassifier(
        cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'
    )

    orientations = [
        (image, "0°"),
        (cv2.rotate(image, cv2.ROTATE_180), "180°")
    ]

    best_orientation = None
    best_score = -1
    best_image = None

    for img, angle in orientations:
        # YOLO детекция с conf=0.4
        results = self.fields_model.predict(img, imgsz=512, conf=0.4,
        verbose=False)[0]
        boxes = results.boxes

        if boxes is not None and len(boxes) > 0:
            num_fields = len(boxes)
            avg_conf = boxes.conf.mean().item()

            # Проверка наличия лица через YOLO
            face_detected_yolo = False
            for i in range(len(boxes)):
                cls = int(boxes.cls[i].item()) if hasattr(boxes.cls[i],
                'item') else int(boxes.cls[i])
                if cls == 2: # класс 2 - это лицо
                    face_detected_yolo = True
                    break

            # Проверка лица через OpenCV

```

```

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(
            gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30)
        )
        face_detected_cv = len(faces) > 0

        # Комплексная оценка
        score = num_fields * 10 + avg_conf * 100
        if face_detected_yolo:
            score += 50
        if face_detected_cv:
            score += 30

        # Минус за отсутствие лица
        if not face_detected_yolo and not face_detected_cv:
            score -= 50

        if score > best_score:
            best_score = score
            best_orientation = angle
            best_image = img

    if best_image is not None:
        return best_image
    else:
        return image

```

Листинг А.4 Реализация гибридной системы распознавания символов (passport_processor.py)

```

def _recognize_field_text(self, image: np.ndarray, field_type: str) ->
Dict[str, any]:
    """Распознавание текста с приоритетом простого Tesseract"""

    # Конвертируем в градации серого если нужно
    if len(image.shape) == 3:
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        gray = image.copy()

    # Сначала пробуем без предобработки для Tesseract
    tesseract_config = self._get_tesseract_config_simple(field_type)
    tesseract_text_raw = ""

    try:
        # Пробуем без обработки
        tesseract_text_raw = pytesseract.image_to_string(gray,
            config=tesseract_config).strip()

        # Очистка текста
        tesseract_text_clean = self._clean_text_simple(tesseract_text_raw,
            field_type)

        # Проверка на валидность
        if self._is_text_valid_simple(tesseract_text_clean, field_type):
            return {
                'text': tesseract_text_clean,
                'confidence': 0.95,
                'method': 'tesseract_raw'
            }
    except:

```

```

        pass

    # Если простой метод не сработал, пробуем с обработкой
    processed = self._preprocess_for_ocr_minimal(gray, field_type)

    # Tesseract с обработкой
    tesseract_text = ""
    tesseract_valid = False

    try:
        tesseract_text = pytesseract.image_to_string(processed,
config=tesseract_config).strip()
        tesseract_text = self._clean_text(tesseract_text, field_type)
        tesseract_valid = self._is_text_valid_for_ocr_switch(tesseract_text,
field_type)
    except:
        pass

    # EasyOCR как запасной вариант
    easyocr_text = ""
    easyocr_conf = 0.0
    easyocr_valid = False

    try:
        if len(processed.shape) == 2:
            processed_rgb = cv2.cvtColor(processed, cv2.COLOR_GRAY2RGB)
        else:
            processed_rgb = processed

        easyocr_results = self.finetuned_easyocr.readtext(processed_rgb)

        if easyocr_results:
            # Берем результат с наибольшей confidence
            best_result = max(easyocr_results, key=lambda x: x[2])
            easyocr_text = best_result[1]
            easyocr_conf = best_result[2]

            # Очистка текста
            easyocr_text = self._clean_text(easyocr_text, field_type)
            easyocr_valid = self._is_text_valid_for_ocr_switch(easyocr_text,
field_type)
    except:
        pass

    # Выбираем лучший результат
    if tesseract_valid and len(tesseract_text) > 3:
        return {
            'text': tesseract_text,
            'confidence': 0.9,
            'method': 'tesseract'
        }
    elif easyocr_valid and easyocr_conf > 0.5:
        return {
            'text': easyocr_text,
            'confidence': easyocr_conf,
            'method': 'finetuned_easyocr'
        }
    elif tesseract_text and len(tesseract_text) >= len(easyocr_text):
        return {
            'text': tesseract_text,
            'confidence': 0.7,
            'method': 'tesseract'
        }
    }

```

```

elif easyocr_text:
    return {
        'text': easyocr_text,
        'confidence': easyocr_conf,
        'method': 'finetuned_easyocr'
    }

return {'text': '', 'confidence': 0.0, 'method': 'none'}

```

Листинг А.5 Реализация интеллектуального fallback-механизма для пустых полей (passport_processor.py)

```

def _smart_ocr_fallback_limited(self, image: np.ndarray, result:
PassportData,
                                fields_data: Dict[str, Dict]):
    """fallback для пустых полей"""

    # Список полей для проверки
    text_fields = ['surname', 'name', 'patronymic', 'birth_date',
'birth_place']

    # Проверка какие поля пустые или некорректные
    empty_fields = []
    for field in text_fields:
        value = getattr(result, field, '')
        if not value or not self.is_text_valid_for_ocr_switch(value, field):
            empty_fields.append(field)
        # Специальная проверка для места рождения
        elif field == 'birth_place':
            # Проверяем на совпадение с ФИО
            if value in [result.surname, result.name, result.patronymic]:
                empty_fields.append(field)
            # Проверяем на запрещенные слова
            elif value in self.validators['birth_place']['forbidden']:
                empty_fields.append(field)

    if not empty_fields:
        return

    # Для каждого пустого поля ищем в области
    for field in empty_fields:
        if field not in fields_data:
            continue

        # Получаем область поля
        field_bbox = fields_data[field].get('bbox')
        if not field_bbox:
            continue

        x1, y1, x2, y2 = field_bbox

        # Расширяем область поиска на 20%
        width = x2 - x1
        height = y2 - y1
        expand_x = int(width * 0.2)
        expand_y = int(height * 0.2)

        x1 = max(0, x1 - expand_x)
        y1 = max(0, y1 - expand_y)
        x2 = min(image.shape[1], x2 + expand_x)
        y2 = min(image.shape[0], y2 + expand_y)

```

```

# Вырезаем расширенную область
search_area = image[y1:y2, x1:x2]

try:
    if len(search_area.shape) == 2:
        search_area_rgb = cv2.cvtColor(search_area,
cv2.COLOR_GRAY2RGB)
    else:
        search_area_rgb = cv2.cvtColor(search_area,
cv2.COLOR_BGR2RGB)

    # OCR только в этой области
    ocr_results = self.finetuned_easyocr.readtext(search_area_rgb)

    best_candidate = None
    best_score = 0

    for bbox, text, conf in ocr_results:
        if conf < 0.3:
            continue

        cleaned_text = self._clean_text_simple(text, field)

        if not cleaned_text:
            continue

        # Проверяем валидность для поля
        if not self._is_text_valid_simple(cleaned_text, field):
            continue

        # Проверяем что текст не является запрещенным словом
        validator = self.validators.get(field, {})
        if cleaned_text in validator.get('forbidden', []):
            continue

        # Специальные проверки для места рождения
        if field == 'birth_place':
            # Не должно совпадать с ФИО
            if cleaned_text in [result.surname, result.name,
result.patronymic]:
                continue
            # Не должно быть датой
            if re.match(r'^\d{2}\.\d{2}\.\d{4}$', cleaned_text):
                continue

        # Оценка
        score = conf

        # Дополнительные проверки для конкретных полей
        if field == 'patronymic' and 'required_endings' in validator:
            if any(cleaned_text.endswith(ending) for ending in
validator['required_endings']):
                score *= 1.5

        if score > best_score:
            best_score = score
            best_candidate = cleaned_text

    # Применяем найденное значение
    if best_candidate and best_score > 0.5:
        setattr(result, field, best_candidate)
        result.confidence_scores[field] = best_score

```



```

        except Exception as e:
            result.processing_errors.append(f"Smart OCR Fallback error for
{field}: {str(e)}")

```

Листинг А.6 Инициализация окна графического интерфейса (passport_gui.py)

```

class PassportApp(QMainWindow):
    """Главное окно приложения"""

    def __init__(self):
        super().__init__()
        self.processor = None
        self.processing_thread = None
        self.images_to_process = []
        self.results_data = []
        self.comparison_db = None
        self.image_delegate = ImageDelegate()

        self.init_ui()
        self.load_processor()

    def init_ui(self):
        """Инициализация интерфейса"""
        self.setWindowTitle("Обработка паспортов РФ")
        self.setWindowIcon(QIcon("icon.png"))

        # Размер окна
        screen = QApplication.desktop().screenGeometry()
        self.resize(int(screen.width() * 0.8), int(screen.height() * 0.8))

        # Центрируем окно
        self.move(int(screen.width() * 0.1), int(screen.height() * 0.1))

        # Центральный виджет
        central_widget = QWidget()
        self.setCentralWidget(central_widget)

        # Главный layout
        main_layout = QVBoxLayout(central_widget)

        # Меню
        self.create_menu()

        # Панель инструментов
        self.create_toolbar()

        # Создаем основную область
        self.create_main_area(main_layout)

        # Создание статусбара
        self.create_statusbar()

```

Листинг А.7 Реализация многопоточной обработки изображений (passport_gui.py)

```

class ProcessingThread(QThread):
    """Поток для обработки изображений"""

```

```

progress = pyqtSignal(int)
result_ready = pyqtSignal(object)
finished = pyqtSignal()
error = pyqtSignal(str)

def __init__(self, processor, images):
    super().__init__()
    self.processor = processor
    self.images = images
    self.is_running = True

def run(self):
    """Запуск обработки"""
    total = len(self.images)

    for i, image_path in enumerate(self.images):
        if not self.is_running:
            break

        try:
            result = self.processor.process_image(image_path)
            self.result_ready.emit(result)
        except Exception as e:
            self.error.emit(f"Ошибка при обработке {image_path}: {str(e)}")

        progress = int((i + 1) / total * 100)
        self.progress.emit(progress)

    self.finished.emit()

def stop(self):
    """Остановка обработки"""
    self.is_running = False

```

Листинг А.8 Реализация экспорта данных в формат Excel

(passport_gui.py)

```

@staticmethod
def export_to_excel(data: List[Dict], output_path: str, selected_fields:
List[str]):
    """Экспорт в Excel"""
    # Подготовка данных для DataFrame
    df_data = []

    for item in data:
        row = {}
        for field in selected_fields:
            if field == 'file_name':
                # Для поля file_name используем прямое название
                row['Файл'] = item.get(field, '')
            elif field in ['original_image', 'page2_image', 'face_image']:
                # Для изображений сохраняется путь или пометка
                row[Config.TABLE_FIELDS[field]] = 'Изображение' if field in
item and item[field] is not None else ''
            else:
                row[Config.TABLE_FIELDS[field]] = item.get(field, '')
        df_data.append(row)

    # Создание DataFrame и сохранение
    df = pd.DataFrame(df_data)

```

```
df.to_excel(output_path, index=False, engine='openpyxl')
```